



**SAVONIA**

# **Lämpötilojen seurantajärjestelmä**

**Jesse Nissinen**

Opinnäytetyö

---



Koulutusala Tekniikan ja liikenteen ala			
Koulutusohjelma Elektroniikan koulutusohjelma			
Työn tekijä(t) Jesse Nissinen			
Työn nimi Lämpötilojen seurantajärjestelmä			
Päiväys	10.4.2012	Sivumäärä/Liitteet	62/10
Ohjaaja(t) yliopettaja Väinö Maksimainen			
Toimeksiantaja/Yhteistyökumppani(t) Työterveyslaitos Kuopio			
<p>Tiivistelmä</p> <p>Opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa lämpötilojen seurantajärjestelmä. Järjestelmä kehitettiin Työterveyslaitokselle Kuopioon.</p> <p>Tarkoituksena oli mitata ja arkistoida mikrobiologian laboratoriossa olevien lämpökaappien sekä laboratorion välittömässä läheisyydessä olevien kylmäsäilytystilojen lämpötiloja. Kaapeissa kasvatetaan erilaisia mikrobeja, joiden vaikutusta tutkitaan eri työpaikkojen terveyshaittojen aiheuttajina. Pakastimet ovat näytteiden säilytystä varten. Aiemmin seuranta on hoidettu kirjaamalla paperille lämpötilat, jotka näkyvät kirjaamishetkellä kunkin mittauskohteen paneelissa.</p> <p>Järjestelmän tuli mitata lämpötiloja säännöllisesti sekä tallentaa ne tietokoneen kovalevyille myöhempää tarkastelua ja arkistointia varten. Tilaaja halusi järjestelmään myös Excel-käyttöliittymän, jonka avulla voitaisiin tehdä analyysia mitatusta datasta mm. grafiikan avulla. Järjestelmän osana haluttiin käyttää jo olemassa olevaa kannettavaa tietokonetta.</p> <p>Järjestelmää varten suunniteltiin tarvittava elektroniikka sekä ohjelmisto PC:lle ja käytettävälle mittalaitteelle. Laitteen prototyyppiä testattiin sen lopullisessa käyttöympäristössä useita kuukausia ennen varsinaista käyttöönottoa. Testien aikana laitteistolla mitattiin jatkuvasti lämpötiloja sekä tarkkailtiin koko järjestelmän toimintaa. Testien sekä asiakkaan palautteen perusteella tehtiin tarvittavat korjaukset, jotta kokonaisuudesta saatiin toimiva.</p> <p>Tuloksena saatiin valmis kokonaisuus, joka täyttää edellä mainitut vaatimukset ja on luotettava vuosiksi eteenpäin. Lisäominaisuutena laitteeseen rakennettiin virheen tunnistus, jonka avulla voidaan havaita mittausväylällä olevien antureiden mahdolliset vikaantumiset.</p>			
Avainsanat lämpötilojen mittaus, digitaalinen, 1-Wire-tekniikka, DS18S20			
julkinen			

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Electronic Engineering			
Author(s) Jesse Nissinen			
Title of Thesis Temperature Monitoring System			
Date	10 April 2012	Pages/Appendices	62/10
Supervisor(s) Mr. Väino Maksimainen, Principal Lecturer			
Project/Partners Finnish Institute of Occupational Health, Kuopio Regional Office			
<p>Abstract</p> <p>The aim of this thesis was to design and build a temperature monitoring system. The system was developed for Finnish Institute of Occupational Health, Kuopio Regional Office.</p> <p>The purpose of this system is to measure and archive the temperatures of cooled incubators in the laboratory of microbiology. The system also had to monitor the temperatures of cold-storage rooms next to this laboratory. Different microbes are grown up in those incubators, and their harmful effects on human health in different working places are researched. Cold-storage rooms are for storing important samples. Earlier this monitoring was done by writing down the temperatures checked out from the front panel of each incubator.</p> <p>The system had to be able to measure temperatures constantly and to save the data on the hard-disk for later examination and archiving. The commissioner also wanted an Excel user interface which could be a helpful tool when analyzing the data for example with graphics. An existing laptop was desired to be used as a part of the system.</p> <p>The needed electronics and the programs for the measuring device and PC were designed. The prototype was tested in its final environment for several months before actual deployment. Temperatures were measured and the whole system was monitored continuously during this test period. The needed improvements were done according to the information got from tests and the feedback got from the customer.</p> <p>As a result of the thesis a completed system meeting the requirements mentioned above and being reliable for years from now on was built. The error-recognition functionality was built as an extra feature, which helps the users to detect possible problems of sensors in the measurement bus.</p>			
<p>Keywords</p> <p>temperature measurement, digital, 1-Wire-technique, DS18S20</p>			
public			

## ALKUSANAT

Kehitin opinnäytetyönäni lämpötilojen seurantajärjestelmän Työterveyslaitokselle Kuopioon. Työ tehtiin vuoden 2011 tammikuun ja vuoden 2012 huhtikuun välisenä aikana. Opin hyvin paljon sulautettujen järjestelmien suunnittelusta sekä tuotekehityksestä. Elektroniikka-alalla niiden molempien hallitseminen on hyvin tärkeää. Tulevaisuuttani ajatellen työ oli minulle erittäin hyödyllinen.

Kiitän opinnäytetyön ohjaajaani yliopettaja Väinö Maksimaista, yhteyshenkilöäni Kari Ojasta ja kaikkia niitä, jotka ovat omalta osaltaan olleet projektissa mukana ja auttaneet työn valmistumisessa.

Kuopiossa 3.4.2012

Jesse Nissinen



## SISÄLTÖ

1	JOHDANTO.....	11
2	MITTAUSYMPÄRISTÖ.....	12
2.1	Lämpökaapit.....	12
2.1.1	Sanyo MIR-253 .....	12
2.1.2	Sanyo MIR-262 .....	13
2.1.3	Memmert ICP-500 .....	14
2.1.4	Memmert UL-50 .....	14
3	MITTAUSTEKNIIKAN SEKÄ LAITTEISTON VALINTA .....	15
3.1	Anturiteknikoiden vertailu.....	15
3.2	Tekniikan valinta.....	15
3.3	Antureiden ohjausyksikkö .....	16
4	1-WIRE.....	17
4.1	Tekniikan erikoisuus .....	17
4.2	Väylän signaalit ja loogiset tasot .....	18
4.3	Ylösvetovastus .....	18
4.4	Liikennöinti 1-Wire-väylällä .....	19
5	MITTAUSLAITTEISTO .....	22
5.1	Arduino Duemilanove .....	22
5.1.1	Digitaaliset I/O-pinnit .....	23
5.1.2	Timer.....	24
5.1.3	Timer/Counter 1 .....	25
5.1.4	USART .....	26
5.2	Digitaalinen lämpötila-anturi DS18S20.....	27
5.2.1	Sähköiset ominaisuudet ja pinnimäärytykset .....	28
5.2.2	DS18S20-anturin protokolla.....	29
5.2.3	Anturin toiminta .....	31
5.2.4	Virransyöttö .....	32
5.2.5	64-bittinen ROM-koodi.....	34
5.2.6	Muisti.....	34
5.2.7	CRC-koodi.....	35
6	ANTURIVERKKO .....	36
6.1	CAT5 UTP -kaapeli.....	36
6.2	Haaroitus.....	37
6.3	Verkon terminologiaa.....	38
6.4	Lineaarinen verkkotopologia .....	39

7	PC:N JA MIKROKONTROLLERIN VÄLINEN RAJAPINTA .....	40
7.1	RS-232 .....	40
7.2	Tiedonsiirto laitteiden välillä .....	40
7.3	Luotettavan tiedonsiirron toteuttaminen .....	41
8	LAITTEISTON RAKENTAMINEN .....	43
8.1	Johdotukset ja kytkennät mikrokontrollerissa .....	43
8.2	Mikrokontrollerin kotelointi ja kotelon muotoilu .....	45
8.3	Antureiden juottaminen ja suojaaminen .....	46
8.4	Anturiverkon haaroitus ja kaapelointi .....	48
9	KÄYTTÖLIITTYMÄN VAATIMUKSET .....	49
10	OHJELMOINTI .....	50
10.1	Mittalaitteen ohjelma .....	50
10.2	PC:n ohjelma.....	54
10.2.1	Ohjelman runko .....	54
10.2.2	Datan tallennus.....	55
10.3	Excel-käyttöliittymä.....	56
10.4	Toiminnan automatisointi ja varmuuskopiointi .....	58
11	YHTEENVETO JA JATKOKEHITYSMAHDOLLISUUDET .....	59
	LÄHTEET.....	61

## LIITTEET

- Liite 1 Pohjakuva mittausympäristöstä
- Liite 2 Mittauslaitteiston käyttöohje
- Liite 3 Bill Of Materials



## LYHENTEET JA KÄSITTEET

1-Wire	Dallas Semiconductors Inc.:n kehittämä latteiden välinen kommunikointiväylä
Microlan	Nimitys 1-Wire-laitteiden muodostamalle verkolle
Ylösvetovastus	Vastus, joka pitää elektronisen komponentin tuloliitännän ylätilassa silloin, kun siihen ei ole kytketty muita signaaleja
I/O-pinni	Input/Output-pinni. Voidaan käyttää sisään- ja ulostuloina
Reset	Aloittaa laitteen ohjelmakoodin suorittamisen alusta
Arduino Duemilanove	Atmega328P-prosessorilla varustettu kehitysalusta sulautetuille järjestelmille
DS18S20	Dallas Semiconductors Inc.:n valmistama digitaalinen lämpötila-anturi
Timer/Counter	Sulautetuissa järjestelmissä käytetty Ajastin/laskuri tapahtumien ajastamiseen ja laskemiseen
USB	(Universal Serial Bus) Sarjaväyläarkkitehtuuri oheislaitteiden liittämiseksi tietokoneeseen
RS232	(Recommended Standard 232) Kahden tietokonelaitteen väliseen kommunikointiin tarkoitettu standardi
FTDI FT232RL	Tekniikka, joka kanavoi sarjaliikennetasoiset signaalit sopivaksi USB-väylälle
Atmega328P	Atmelin valmistama sulautettujen laitteiden prosessorimalli
SRAM	(Static Random Access Memory) Puolijohdetekniikalla toteutettu muistityyppi
EEPROM	(Electrically Erasable Programmable Read-Only Memory) Haihtumaton puolijohdetekniikalla toteutettu muistityyppi
AVR	Atmelin 8-bittisten mikrokontrollerien tuoteperhe
A/D-muunnos	Analogisen signaalin muuntaminen digitaalseksi
Scratchpad	DS18S20-anturin muistilohko, josta lämpötiladata luetaan
Cat5 UTP	(Category5 Unshielded Twisted Pair) Parikaapelityyppi, jota käytetään mm. Ethernet-verkoissa
RJ45	Lähiverkkojen yleisin liityntätyyppi parikaapeleille
LED	(Light Emitting Diode) Puolijohdekomponentti, joka tuottaa valosäteilyä

GND	Lyhenne sanasta ground, joka tarkoittaa elektroniikassa maatasoa (0 voltia)
VBA	(Visual Basic for Applications) Ohjelmointikieli, joka sisältyy mm. Microsoft Officeen
Isäntä/Master	(Väyläisäntä) Kommunikointiväylän laite, jonka käskystä muut laitteet suorittavat halutut toiminnot
Orja/Slave	(Väyläorja) Kommunikointiväylän laite, joka suorittaa väyläisännän käskemät toiminnot

## 1 JOHDANTO

Kehitetty seurantajärjestelmä tilattiin Kuopion Työterveyslaitoksen laboratorioon ja sen välittömässä läheisyydessä sijaitseviin kylmäsäilytystiloihin. Laboratoriossa kasvatetaan eri työpaikkojen työympäristöistä otettuja näytteitä, joissa epäillään olevan ilmanlaatuun ja ihmisten terveyteen vaikuttavia mikrobeja. Näytteitä pidetään kasvun mahdollistavalla kasvualustalla, ja erilaisten näytteiden viljely edellyttää kullekin lajille sopivaa lämpötilaa. Tämän kasvualustan ja lämpötilan tarjoavat laboratorion lämpökaapit. Riittävän viljelyn jälkeen lajit tunnistetaan mikroskooppisesti. Kaappien lämpötilojen seuraaminen on tärkeää, sillä suurista kaapeissa tapahtuneista lämpötilanvaihteluista voi olla haitallisia seurauksia näytteille. Mittauskohteen toinen segmentti on laboratorion välittömässä läheisyydessä sijaitsevat kylmäsäilytystilat, joiden lämpötilojen stabiilius on myös tärkeää. Sattuneet vikatilanteet ja niiden ajankohta voidaan tarkastaa kehitetyn järjestelmän laatimasta lämpötilaraportista. Kaikkien mittauspisteiden lämpötilanseuranta on tähän mennessä toteutettu kirjaamalla lämpötilat kynää ja paperia käyttäen. Tämä ei ole erityisen kätevää, eikä seuranta ole ollut niin tiivistä, kuin sen ehkä olisi hyvä olla. Tämän vuoksi tarvittiin automaattinen ja itsenäisesti toimiva järjestelmä, joka mittaa ja kerää lämpötiloja säännöllisesti.

Opinnäytetyössä kehitetään mittauslaitteisto, joka on kustannustehokas sekä luotettava. Laite helpottaa laboratorion henkilökuntaa, eikä seurantaan jää välipäiviä vapaapäivien tai lomien vuoksi. Fyysisesti laitteisto koostuu kannettavasta tietokoneesta sekä siihen liitetystä anturiverkosta rajapintoineen. Toinen puoli kokonaisuudesta on ohjelmisto, joka kehitettiin toteuttamaan laitteiston toiminnalliset vaatimukset. Järjestelmä suorittaa mittaukset tasaisin väliajoin ja tallentaa ne tietokoneen kovalevylle tarkastelua ja myöhempää analysointia varten.

Raportissa pyritään selvittämään käytettävien tekniikoiden periaatteet siten, että niiden pääasialliset toimintaperiaatteet tulevat tutuiksi. Suuri osa työhön käytetystä ajasta kului itse laitteiston rakentamiseen sekä ohjelmiston kehitykseen.

## 2 MITTAUSYMPÄRISTÖ

Mittausympäristö käsittää kokonaisuudessaan 12 mittauspistettä. Näistä seitsemän on mikrobiologian laboratoriossa sijaitsevia lämpökaappeja ja loput viisi laboratorion välittömässä läheisyydessä olevia kylmäsäilytystiloja. Kylmäsäilytystilat ovat isoja huoneita, joista yksi on pakastin ja loput jääkaappeja. Työssä ei perehdytä kylmäsäilytystiloihin tämän tarkemmin. Erilaisten mittauspisteiden vuoksi mitattavat lämpötilat vaihtelevat pakkasesta lämpimään. Lämpötila-alue on noin  $-20\text{ °C} \dots +55\text{ °C}$ .

### 2.1 Lämpökaapit

Laboratoriossa on neljä erimallista lämpökaappia. Osa kaapeista on hankittu yli 20 vuotta sitten ja osa kymmenen vuoden sisällä. Tämän vuoksi ne vaihtelevat ominaisuuksiltaan. Jokaisen luvun lopussa on kuva kyseisessä luvussa esitellystä kaapista.

#### 2.1.1 Sanyo MIR-253

MIR-253-kaappeja (kuva 1) on tässä laboratoriossa neljä kappaletta. MIR-sarjan kaapeissa on 8-bittinen mikroprosessori, jolla ohjataan lämmitystä ja jäähdytystä  $\pm 0,2\text{ °C}$ :n tarkkuudella (Sanyo 2012). Muita MIR-253:n tärkeitä ominaisuuksia ovat mm:

- P.I.D-ohjaus sekä kompressorin On-Off-ominaisuus
- myötäkytkentätoiminto kompressorin oikea-aikaiseen toimintaan
- lämpötilan laaja asetusalue  $-20\text{ °C} \dots +50\text{ °C}$
- pieni lämpötilanvaihtelu kaapin sisällä  $\pm 0,5\text{ °C}$
- suuri kapasiteetti 254 l
- käyttömahdollisuudet eri aloilla, esimerkiksi mikrobiologiassa sekä elintarviketeollisuudessa.

(Sanyo 2012)



KUVA 1. Sanyo MIR-253 (Sanyo 2012)

Sanyo MIR-262:ssa (kuva 2) on pääpiirteittäin samat ominaisuudet kuin kooltaan suuremmassa MIR-253:ssa. Oleelliset erot ovat pienempi kapasiteetti (126 litraa) sekä asetettava lämpötila-alue (5 °C...+80 °C) (Sanyo 2012). Laboratoriossa ainoastaan tämän kaapin lämpötila on yli 50 °C:n. Mainitaan lisäksi joitakin kaapin käyttöön liittyviä ominaisuuksia:

- lämpötilanvaihtelu kaapin sisällä  $\pm 0,5$  °C
- kosketusnäytöllinen ohjauspaneeli
- mikroprosessoriohjattu
- ajastetut ohjelmat.

(Sanyo 2012)



KUVA 2. Sanyo MIR-262 (Sanyo 2012)

Kuvassa 2 oleva MIR-262 on huomattavasti pienempi kuin MIR-253.

### 2.1.3 Memmert ICP-500

Memmert ICP-500 (kuva 3) on ICP-sarjan 108-litrainen jäähdytetty lämpökaappi. Tarkan ohjaustekniikkansa ansiosta lämpötilojen asetusravot saavutetaan ilman energiaa kuluttavia tehopiikkejä. Nopeat lämmitys- ja jäähdytysajat tekevät tästäkin mallista erinomaisen valinnan mikrobiologian laboratorioon (Memmert 2012). ICP-500-kaapin olennaisia toiminnallisia ominaisuuksia ovat:

- mikroprosessoriohjattu P.I.D-säätö
- lämpötila-alue 0 °C...+60 °C
- lämpötilanvaihtelu <0.3 °C alueella +10 °C...+37 °C
- asettavat hälytysrajat minimi- ja maksimilämpötiloille.

(Memmert 2012)



KUVA 3. Memmert ICP-500 (Memmert 2012)

### 2.1.4 Memmert UL-50

Memmert UL-50 (kuva 4) on kaapeista selvästi vanhin. Lämmitysominaisuuksiltaan se on erilainen verrattuna muihin laboratoriossa käytettyihin lämpökaappeihin. Lämpötila-alue on 30 °C...+220 °C, eikä siinä ole digitaalista käyttöpaneelia.



KUVA 4. Memmert UL-50 (Memmert 2012)

### 3 MITTAUSTEKNIIKAN SEKÄ LAITTEISTON VALINTA

Ennen varsinaisen työn aloittamista tuli valita olosuhteisiin sopiva mittaustekniikka ja siihen soveltuva laitteisto. Koska työn tilaajalla, itselläni tai ohjaavalla opettajalla ei ollut yhtä tiettyä kokonaisuutta valmiina toteutettavaksi, olivat eri vaihtoehdot avoinna. Aluksi eräänä mahdollisuutena mieleen tulivat langattomat lämpötila-anturit, jotka olisi vaivatonta sijoittaa mittauspisteisiin ja tarvittaessa niitä olisi helppo huoltaa tai muuttaa niiden sijaintia mittaussivkossa. Vaihtoehtona olivat luonnollisesti galvaanista yhteyttä käyttävät langalliset anturit.

#### 3.1 Anturitekniikoiden vertailu

Kartoituksen aikana vastaan tuli kuitenkin langattomien tekniikoiden ominaisuuksia, jotka eivät tässä tilanteessa soveltuneet tilaajalle parhaalla mahdollisella tavalla. Näistä mainittakoon korkeahkot kustannukset sekä monien langattomien tekniikoiden soveltumattomuus tuleviin tiloihin, joissa mittauspisteet ovat. Jotkin mittauspisteet ovat paksun betoniseinän takana ja jopa pommisuojusta, joten signaalien kuuluvuus ei olisi riittävä tai se katkeaisi mahdollisesti kokonaan. Kun langattomien antureiden käyttö oli suljettu pois, jäljelle jäivät langalliset, fyysisellä yhteydellä toimivat anturit. Tilanne kääntyi käytännössä pääläelleen. Antureiden sijoittelu ei tulisi olemaan niin yksinkertaista kuin langattomien antureiden, mutta elektromekaniikka olisi yksinkertaisempaa, kustannukset tippuisivat murto-osaan eikä kuuluvuusongelmia olisi lainkaan.

#### 3.2 Tekniikan valinta

Työssä tuli määritellä se, olivatko anturit toiminnaltaan analogisia vai digitaalisia. Koska antureista ei itselläni ollut käytännön kokemuksia, täytyi aihetta opiskella lukemalla erilaisia artikkeleita. Internetistä löytyi melko paljon tekstejä erilaisista anturintekniikoista. Lueskelin kirjoituksia myös erilaisilta tunnetuilta elektroniikkaan keskittyneiltä keskustelufoorumeilta, jolloin kuva yleisesti käytetyistä lämpötilojen mittaustekniikoista alkoi muodostua. Kaiken kaikkiaan analogisten antureiden suurin etu verrattuna digitaalisiin antureihin oli niiden mittaustarkkuus.

Selvitystyö vei aikaa melko paljon, mutta sen aikana saatiin kartoitettua sopivat vaihtoehtoiset tekniikat. Päällimmäisenä mieleen jäi Dallas Semiconductors Inc:n kehittämä tiedonsiirtotekniikka nimeltä 1-Wire. Alkoi varmistua, että anturit olisivat digitaalisia. Digitaalitekniikka on häiriönsiedollisesti parempi kuin analoginen tekniikka. Lisäksi digitaalisia antureita ei tarvitse kalibroida, mikä helpottaa ohjelmointityötä. 1-Wire-tekniikan parhaina puolina pidin tarvittavien johdinten vähäisyyttä sekä mahdollisuutta käyttää digitaalisia antureita. Lisäksi valintaa tukevia seikkoja olivat tekniikkaan liittyvä helposti saatavilla oleva kattava informaatio sekä oppilaitokselta testiin saatavat 1-Wire-anturit, joita oli mahdollista päästä välittömästi kokeilemaan. Antureille oli saatavilla paljon ohjeita mm. ohjelmointia varten. Lisäksi niiden tarkkuus ja muut toiminnalliset ominaisuudet vastasivat tilaajan tarpeita, joten anturityypin valinta oli helppo.

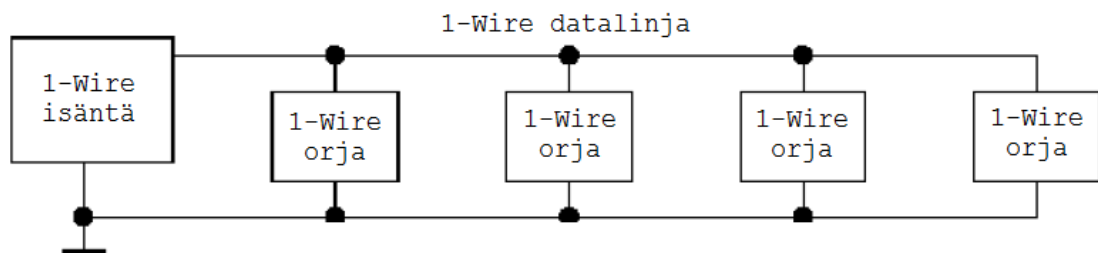
### 3.3 Antureiden ohjausyksikkö

Antureille täytyi vielä valita isäntäohjain, jonka komentamana anturit saadaan suorittamaan mittauksia ja joka toimisi rajapintana mittausverkon ja PC:n välillä. Tähän tarkoitukseen valittiin oppilaitokselta opetuskäytössäkin ollut prosessorikortti, jota oli mahdollista ohjelmoida tutuilla ohjelmankehitysympäristöillä. Prosessorikortin ominaisuudet riittivät mittauslaitteiston toteutusta varten. Mittauskortti, mittaustekniikka ja anturit alkoivat olla käytännössä tilaajan ja ohjaavan opettajan hyväksyntää vaille valittuina. Molemmat osapuolet hyväksyivät valitsemani systeemin, joten pääsin tutustumaan syvemmin kokonaisuuteen.



## 4 1-WIRE

1-Wire-tekniikka on Dallas Semiconductorsin kehittämä tiedonsiirtomenetelmä elektronisten laitteiden välille. Se koostuu kolmesta pääelementistä: 1-Wire-rajapinnasta (väyläisäntä), orjalaitteista sekä kytkennästä isännän ja orjien välillä. 1-Wire-verkkoa kutsutaan Microlaniksi (kuvio 1). Orjat kytketään väylälle rinnakkain.



KUVIO 1. Microlan (Springbok 2004, 4)

1-Wire-tekniikan perustana on sarjaliikenne, joka käyttää ainoastaan yhtä datalinjaa sekä referenssimatkaa kommunikoinnin toteuttamiseen. Väyläisäntä (tässä työssä mikrokontrolleri) hoitaa alustukset ja ohjaa kommunikointia yhden tai useamman väylällä olevan orjalaitteen kanssa. Kaikilla väylän orjalaitteilla on uniikki ja haihtumaton tehtaalla ohjelmoitu 64-bittinen ID-tunnistenumero, joka toimii osoitteena laitteen tunnistamiseen väylällä. 8-bittinen tuoteperhenumero taas osoittaa laitteen tyyppin ja sen toiminnallisuudet. Laitteet toimivat tyypillisesti 2.8 V – 5.25V jännitealueella. Useimmissa 1-Wire-laitteilla ei ole erillistä pinniä käyttöjännitteelle, vaan ne käyttävät ns. loisivirtaa. Tällöin tarvittava energia saadaan datalinjalta. (Linke 2008.)

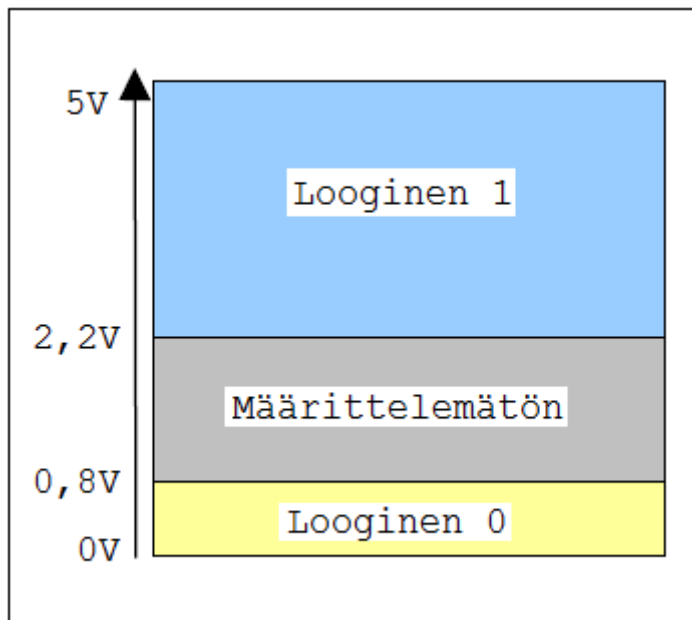
### 4.1 Tekniikan erikoisuus

1-Wire-tekniikka on ainoa jännitteeseen perustuva digitaalinen järjestelmä, joka toimii ainoastaan kahdella kontaktilla: datalinjalla ja maalla (0 V). Tämän ansiosta voidaan käyttää niin kutsuttua kaksisuuntaista (half-duplex) -kommunikointia. Kontrastina muihin sarjaliikennesysteemeihin, kuten I<sup>2</sup>C:hen tai SPI:hin, 1-Wire-laitteet on suunniteltu kontaktiympäristöön. Laitteen kytkeytyessä irti väylästä tai sen kadottaessa kontaktin, asettuu laite määritellyyn reset-tilaan. Kun jännite palaa, laitteet heräävät ja ilmoittavat olemassaolostaan. 1-Wire-laitteiden ESD-suojaus on hyvä. Lisäksi 1-Wire-laitteet ovat taloudellisin tapa lisätä elektronista toimintaa moneen erilaiseen järjestelmään. Sovelluksia voivat olla esimerkiksi tunnistamista ja varmentamista vaativat

kohteet, kuten kulunvalvonta ja henkilökohtaisten tietojen varmentaminen. 1-Wire-laitteilla voidaan myös jakaa kalibrointidataa ja erilaisia valmistustietoja. (Linke 2008.)

#### 4.2 Väylän signaalit ja loogiset tasot

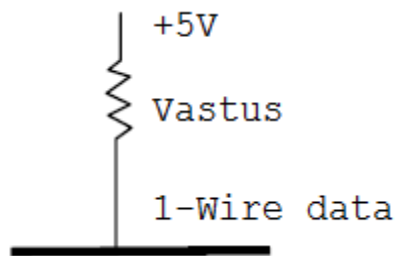
1-Wire-väylän informaatio kulkee digitaalisesti. Sen loogiset tasot ovat karkeasti seuraavanlaiset: keskimääräinen maksimi loogiselle ”nollalle” on 0,8 V ja keskimääräinen minimi loogiselle ”ykköselle” on 2,2 V. Näiden väliin jäävää aluetta kutsutaan ”määrittelemättömäksi” (kuvio 2). Tämän alueen läpi tapahtuvien signaalien siirtymien tulisi olla nopeita, koska alue on loogisesti määrittelemätön. (Springbok 2004, 5.)



KUVIO 2. 1-Wire-väylän loogiset tasot (Springbok 2004, 5)

#### 4.3 Ylös- ja alaspäin vetovastus

Datalinja kytketään tyypillisesti 5 V:n syöttöjännitteeseen vastuksen kautta kuvion 3 mukaisesti. Näin datalinja on jatkuvasti vedettynä kohti loogista ”ykköstä” tämän ylös- ja alaspäin vetovastuksen kautta. Avokollektorityyppisen lähtönsä vuoksi sekä isäntä että orjat voivat oikosulkea datalinjan maatasoon vaihtaakseen loogisen tason 1:stä 0:aan. Kun lähetystä ei tapahdu tai liikennettä ei ole, väylän looginen taso on 1 ( $\geq 2,2$  V). ”Lähetystaso” taas on 0 ( $\leq 0,8$  V). Vastusarvot ovat tyypillisesti 1 k $\Omega$  - 4,7 k $\Omega$ . (Springbok 2004, 6.)

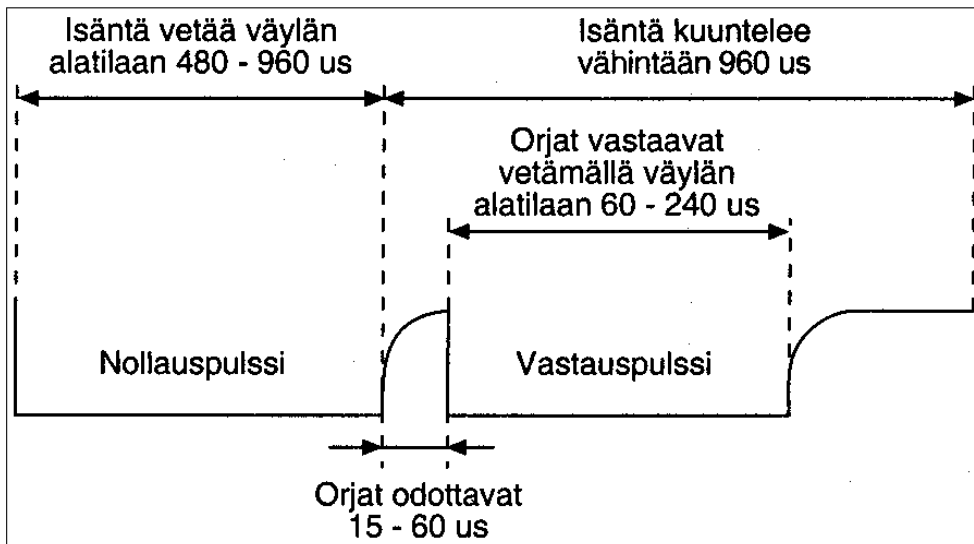


KUVIO 3. Väylän ylös vetovastuksen kytkentä (Springbok 2004, 6)

#### 4.4 Liikennöinti 1-Wire-väylällä

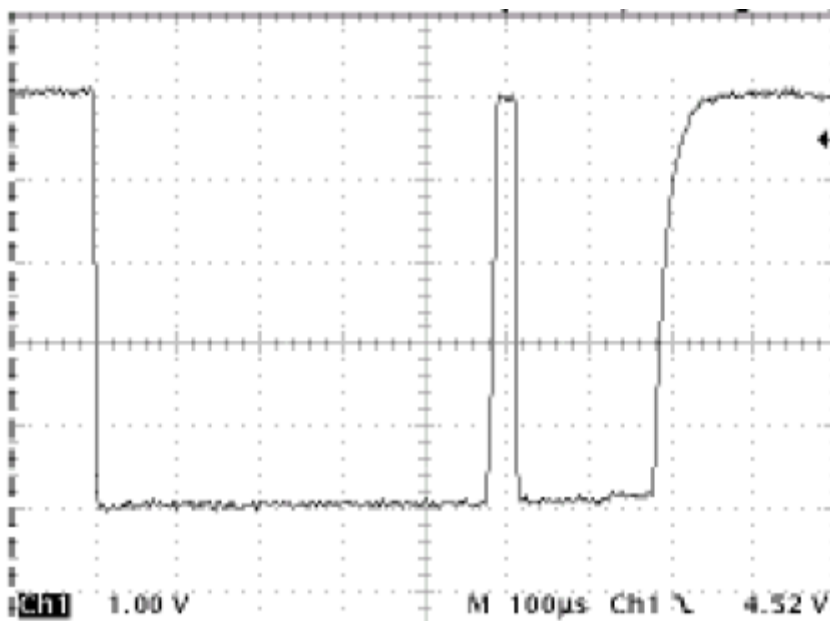
Isäntä on aina se laite, joka aloittaa tiedonsiirron 1-Wire-väylällä. Lepotilassa ollessaan väylä on aina ylätilassa. Komponentit ottavat käyttöjännitteensä edellä mainitun lepotilan aikana, kun ne ovat liitettyinä väylään vain kahdella johtimella (data ja maa). Väylä on alatilassa vain hyvin lyhyitä aikoja, jolloin komponentit saavat käyttöjännitteensä piirin sisäisestä kondensaattorista (Koskinen 2004, 258). Tässä tilanteessa orjalaitteiden sanotaan käyttävän loisvirtaa (parasite power).

Väylään kytkettävien komponenttien ja laitteiden dataliitännän lähtö on avokollektori-tyyppinen, jolloin myös isännän I/O-liitännän lähdön tulee olla samanlainen kuin orjan eli avokollektori-tyyppinen. Tässä tilanteessa jokainen väylään kytketty laite tai komponentti kykenee vetämään väylän tarvittaessa alatilaan. 1-Wire-väylän liikennöinti aloitetaan nollauspulssilla (Reset-pulse) (kuvio 4), jolloin isäntä vetää väylän alatilaan 480...960  $\mu$ s ajaksi. Nollauspulssin jälkeen isäntä kuuntelee väylää ja väylä nousee takaisin ylätilaan. Mikäli väylällä on orjia, ne osoittavat olemassa olonsa vetämällä väylän alatilaan 15...60  $\mu$ s kuluttua 60...240  $\mu$ s ajaksi. Näin isäntä saa tietää, onko väylällä laitteita. (Koskinen 2004, 258.)



KUVIO 4. 1-Wire-väylän liikennöinnin aloitus (Koskinen 2004, 258)

Kuviossa 4 esitetty liikennöinnin aloitus näyttää oskilloskoopilta katsottuna seuraavanlaiselta (kuva 5).

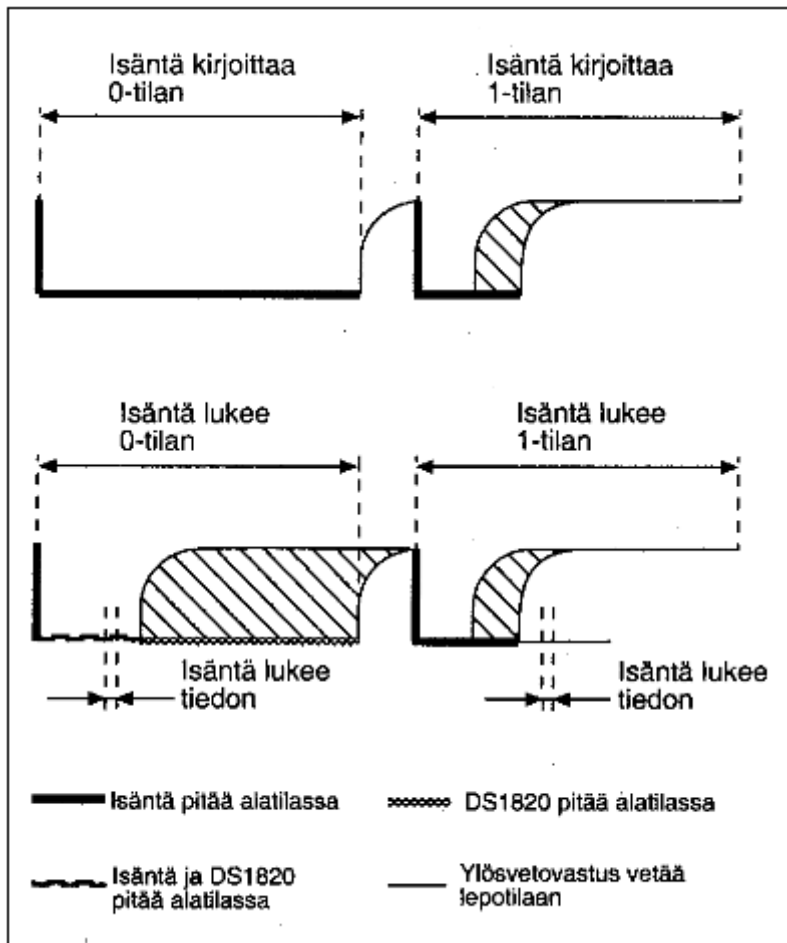


KUVA 5. 1-Wire-väylän liikennöinnin aloitus (Springbok 2004, 8)

Edellä kuvattua tilannetta voidaan käyttää ohjelmallisesti myös etsittäessä väylältä antureita. Sovellettuna se on tapa lisätä isäntälaitteeseen esimerkiksi virheentunnistus; mikäli isäntä ei havaitse väylällä oikeaa määrää oikeita pulsseja, puuttuu väylältä laite tai siihen on kytketty jotain ylimääräistä.

Nollauspulssin jälkeen orjat ovat valmiita kommunikointiin mikrokontrollerin kanssa. Protokollan mukaan vain isäntä voi lukea tai kirjoittaa tietoa väylällä. 1-Wire-väylän

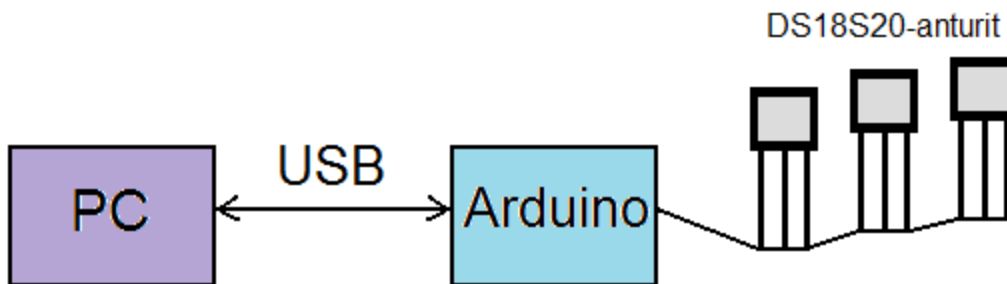
luku- ja kirjoituspulssien aikakaaviot esitetään kuviossa 5, josta voidaan nähdä kirjoituspulssseja olevan kahdenlaisia: Ykkösbitin kirjoitus ja nollabitin kirjoitus. Kirjoitus- ja lukutoimintoihin kuluvia ajanjaksoja kutsutaan aikaikkunoiksi. Kunkin kirjoituspulssin keston täytyy olla vähintään 60  $\mu$ s ja pulssien välissä on oltava vähintään 1  $\mu$ s väli. Isännän vetäessä väylän alatilaa, orja lukee väylän tilan 15...60  $\mu$ s kuluttua. Jos väylä on yhä alatilassa, orja tulkitsee tilan nollassi; jos väylä on palautunut takaisin ylätilaan, orja tulkitsee tilan ykköseksi. Isännän lukiessa tietoa orjasta, aloitetaan lukupulssi vetämällä väylä alatilaa vähintään 1  $\mu$ s ajaksi. Isäntä päästää avokollektori-lähtönsä lepotilaan ja lukee hieman tämän jälkeen väylän tilan. Jos orjasta luetaan nollassi, orja vetää väylän alatilaa heti ja pitää väylää alatilassa vähintään 15  $\mu$ s. Jos orjasta luetaan ykkössi, orja ei vedä väylää alatilaa lukupulssin jälkeen. (Koskinen 2004, 258.)



KUVIO 5. 1-Wire-väylän kirjoitus- ja lukupulssit (Koskinen 2004, 259)

## 5 MITTAUSLAITTEISTO

Mittauslaitteisto (kuvio 6) koostuu Arduino Duemilanove -prosessorikortista sekä siihen kytketyistä Dallas Semiconductors:in digitaalisista DS18S20 -lämpötila-antureista. Laitteistokokonaisuuteen kuuluu myös kannettava tietokone, johon mittausdata lopulta kerätään talteen ja jatkokäsittelyä varten.

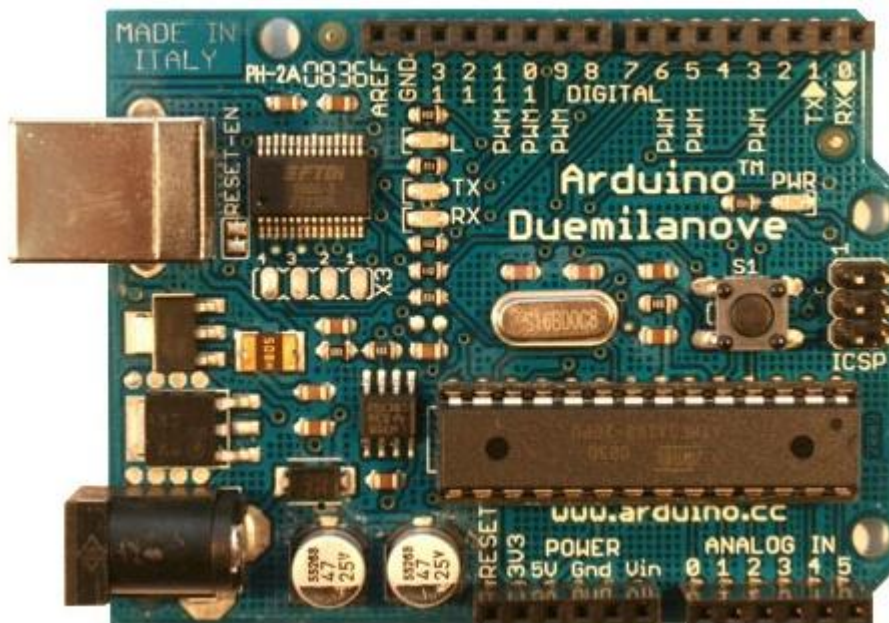


KUVIO 6. Mittauslaitteisto

1-Wire-liittymän olisi voinut toteuttaa suoraan PC:n RS232-portin kautta ilman mikrokontrolleria, mutta kortin mukaan ottaminen tuo järjestelmään paljon etuja. Ensimmäisenä ja suurimpana hyötynä todettiin mahdollisilta virtakatkoksilta ja muilta PC:hen liittyviltä vikatilanteilta suojautuminen. Mittausdata voidaan esitallentaa kortilla olevaan muistiin ja purkaa sieltä vasta halutessa PC:lle. Mikäli PC sammuu tai vikaantuu, ei viimeisimpiä mittautietoja kadoteta virtakatkoksen tai jumiutumisen myötä. Kannettava tietokone toimii samalla ”varavirtana” prosessorikortille, sillä mikäli sähköverkko kaatuu, riittää PC:n akusta virtaa tunneiksi.

### 5.1 Arduino Duemilanove

Arduino Duemilanove (Kuva 6) on mikrokontrollerialusta, jonka perustana on joko ATmega168 tai ATmega328 -prosessori. Tässä työssä on käytetty alustaa, jossa on ATmega328 -prosessori. Kuten kuvasta näkyy, siinä on 14 digitaalista sisääntuloa/ulostuloa ja kuusi analogista sisääntuloa. Alustalla on USB-liitäntä, virtaliitin, reset-nappi sekä kaikki muu tarvittava tuki mikrokontrollerille. Kommunikointi tietokoneen kanssa tapahtuu USB:n välityksellä. Tämä toimii siten, että alustalla oleva FTDI FT232RL -piiri kanavoi sarjaliikenteen signaalit USB-väylälle sopivaksi. Toisin sanoen tiedonsiirto tapahtuu USB-väylää pitkin, mutta väylän ohjelmointi on suoritettava kuten RS232:lle. (Arduino 2011.)



KUVA 6. Arduino Duemilanove (Arduino 2011)

Kootaan yhteenveto alustan ominaisuuksista:

- ATmega328P-prosessori
- toimintajännite 5 V
- syöttöjännite 5 – 12 V
- 14 digitaalista I/O-pinniä
- 6 analogista sisääntulopinniä
- 32 KB:n Flash-ohjelmamuisti
- 2 KB:n SRAM
- 1 KB:n EEPROM
- kellotaajuus 16 MHz.

(Arduino 2011)

### 5.1.1 Digitaaliset I/O-pinnit

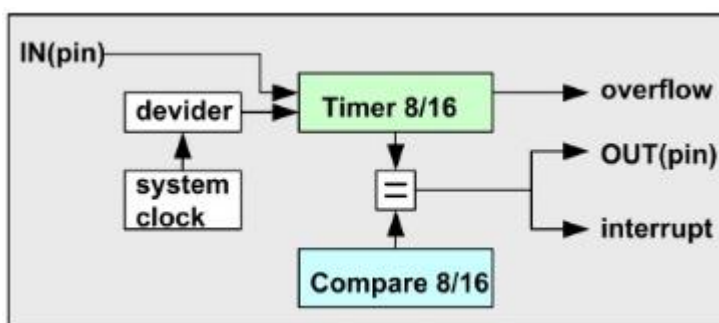
Arduinossa on 14 digitaalista I/O-pinniä. Pinnit voivat toimia konfiguroinnin mukaisesti joko sisään- tai ulostulopinneinä. Ne ovat asetusarvoisesti sisääntuloja, joten jos niitä halutaan käyttää ulostuloina, tulee ne ohjelmallisesti määrittää ulostuloiksi. Atmegan pinnit voivat tarjota 40 mA:n virran, joka on riittävä moniin eri tarkoituksiin. Esimerkiksi työssä käytetyn DS18S20-lämpötila-anturin sähköisten ominaisuuksien perusteella yhteen Atmegan pinniin voisi liittää

$$40 \text{ mA} / 1.5 \text{ mA} = 26,667$$

eli noin 27 anturia. Laskukaavassa 1.5 mA on DS18S20:n käyttämä virran maksimi arvo ja 40 mA on digitaalisen pinnin syöttämä maksimi virta.

### 5.1.2 Timer

Mittauslaitteiston erittäin tärkeä osa on kello, jonka avulla mittaukset saadaan tehtyä juuri oikeaan aikaan. Mikrokontrollerissa tätä varten on omalla ominaistaajuudellaan värähtelevä kide, jota voidaan käyttää tähän tarkoitukseen. Timeriä voidaan kutsua myös ajastimeksi tai laskuriksi. Kuvio 7 esittää timerin lohkokaaviona.



KUVIO 7. Timerin lohkokaavio (Vahtera 2008)

Ajastin voi olla 8- tai 16-bittinen, ja se voi laskea käytettävän prosessorin sähköisiä pulsseja tai ulkoisia tapahtumia. Ajastimen lähtö voi antaa pulsseja suoraan portin pinniin tai generoida keskeytyksen. Periaatteessa ajastimet ovat ohjelmoitavia pulsseja laskevia laskureita, joita on ominaisuuksiltaan erilaisia. Toisissa on perustoiminnon eli ylivuototilan (overflow mode) lisäksi vertailutoiminta (compare mode). Joissain on lisäksi kaappausominaisuus (capture mode) sekä pulssinleveysmodulointi (PWM) monella eri tavalla. AVR-ohjaimissa on lyhyeen laskentaan 8- ja pitkään laskentaan 16-bittiset ajastimet. Useimmissa, kuten ATmega328:ssa, 8-bittisiä on kaksi kappaletta, Timer 0 ja Timer 2 (sis. async toiminnon = oma kello) ja monipuolisin ominaisuuksin varustettu 16-bittinen Timer 1. Jos tarvitaan erityisen pitkää ajastusta, voidaan eri ajastimia kytkeä toimimaan peräkkäin. Pienin aika, jonka ajastimella voi mitata, on kahden laskettavan pulssin välinen aika. (Vahtera 2008.)

Ajastin voidaan asettaa monenlaiseen toimintaan, modeen, ohjausrekistereitä sopivasti manipuloimalla. Toiminnan kannalta tärkeää on ymmärtää se, että laskenta tapahtuu muusta ohjelmakoodista riippumatta. Ajastin toimii itsenäisesti, mutta samalla se tarvittaessa reagoi ulkoisiin oheislaitteisiin ja tapahtumiin ja ohjaa niitä koodin käskyjen määräysten mukaan.

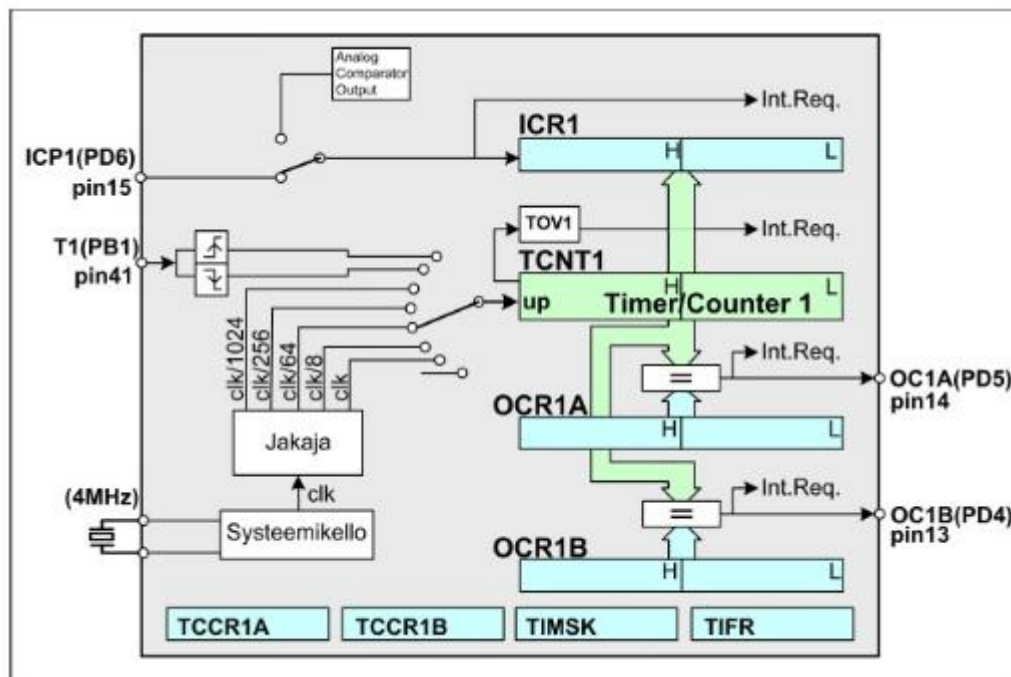


Ajastinta tarvitaan mm.

- ajallisesti tarkkojen viiveiden tekemiseen riippumatta muusta koodista
- erilaisten kellojen (mm. real time clock) tekemiseen
- pulssien, kierrosluvun ja tapahtumien laskentaan
- pulssin leveyden ohjaamiseen, säätämiseen tai mittaamiseen
- mittaamaan aikaa kahden tapahtuman välillä.

### 5.1.3 Timer/Counter 1

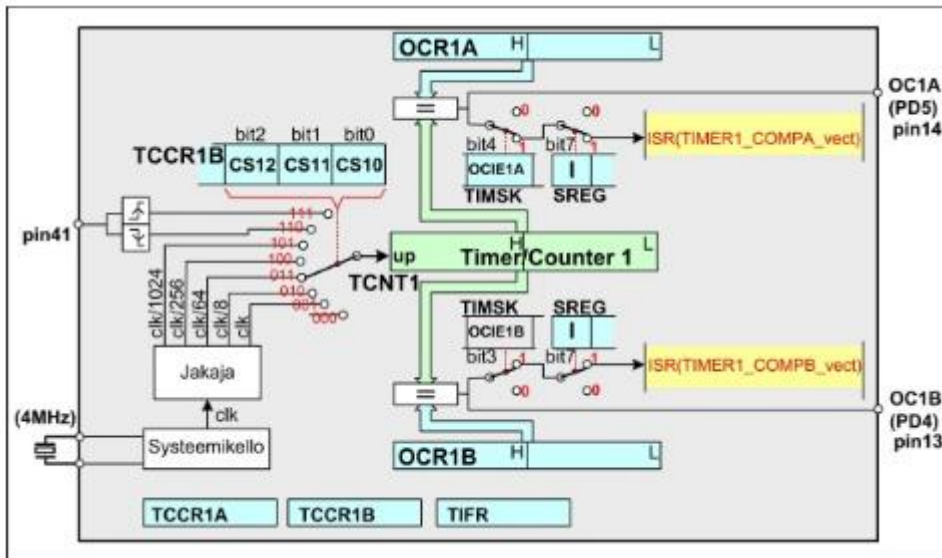
Atmega328P-ohjaimessa ajastintoimintojen ytimenä toimii Timer1 (kuvio 8) ja sen TCNT1-rekisteri, joka koostuu rinnakkain kytketyistä kiikuista. Laskurilla voidaan laskea lukuja nolasta (0x0000) aina lukuun 65535 (0xFFFF) asti. TCNT1 koostuu kahdesta 8-bittisestä rekisteristä, joista käytetään nimitystä "High" ja "Low". Resetoinnin jälkeen laskurin alkuarvo on nolla, siis sen kaikki bitit ovat nollia. Laskenta voidaan aloittaa siitä. Rekisteriä voidaan sekä lukea että siihen voidaan kirjoittaa mikä tahansa 16-bittinen luku. Eli laskennan alkuarvo on ohjelmoijan määritettävissä. Laskenta tapahtuu tavallisesti ylöspäin eli  $0 \rightarrow 65535 = 0xFFFF$ . Kun laskuri on täynnä, siis 0xFFFF, seuraava tuleva pulssi aiheuttaa laskurin ylivuodon. (Vahtera 2008.)



KUVIO 8. Atmega 328P Timer/Counter1 lohkokaavio (Vahtera 2008)

Tässä työssä käytetään Timer1:n vertailutilaa eli Compare Modea sekä ISR-keskeytysfunktioita, joka toimii itsenäisesti riippumatta muusta koodista. Compare Modessa (kuvio 9) verrataan kaiken aikaa TCNT1-rekisterin ja vertailurekisterin OCR1A (tai OCR1B) arvoja. Kun ne ovat yhtä suuret, asettuu TIFR-lippurekisterin

Output Compare Flag seuraavalla kellopulssilla. Jos OutputCompare Interrupt on määritetty (ja globaali keskeytysbitti sen sallii), hypätään keskeytysfunktioon, jossa päivitetään kellonaikaa (Vahtera 2008).



KUVIO 9. Output Compare Mode (Vahtera 2008)

Kuviossa 9 esitetty Output Compare Mode toimii ohjelman taustalla itsenäisesti eikä vie siten prosessoriaikaa. Taulukossa 1 kerrotaan lyhyesti siihen liittyvien rekistereiden käytöstä.

TAULUKKO 1. Timer1:n Output Compare Mode -rekistereiden käyttö

Rekisteri	Käyttö
OCR1A/B	Vertailuarvon asetus
TCCR1B	Toimintatilan ja kellopulssien jakoluvun asetus
TIMSK	Keskeytyksien asettaminen

Esitettyjen rekistereiden oikealla käytöllä ajastin saadaan toimimaan halutulla tavalla.

#### 5.1.4 USART

USART (Universal Synchronous-Asynchronous Receiver/Transmitter) on tärkeä ohjainpiiri tietokoneen ja oheislaitteiden välisessä sarjaliikenteessä. Datatavut lähetetään yksittäisinä bitteinä peräkkäin määränpäähän, jossa ne kootaan takaisin kokonaisiksi tavuiksi. Sarjaliikenteelle on kaksi muotoa: asynkroninen ja synkroninen. Käytettävän muodon määrää laitteisto.

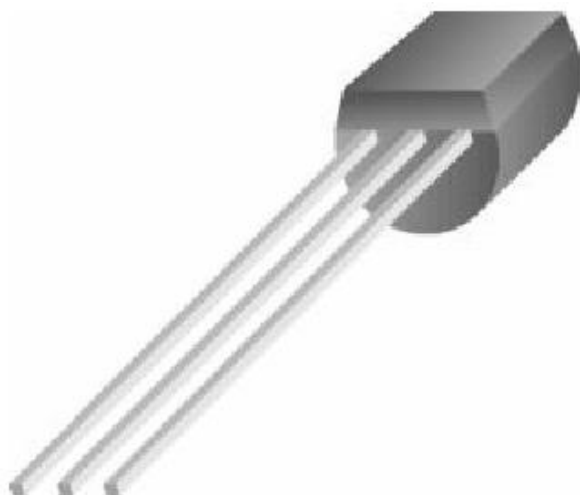
Synkroninen toiminta edellyttää, että molemmat sarjaliikenteen osapuolet jakavat jonkin ajastustavan keskenään, jotta vastaanottaja tietää, milloin lukea seuraava tuleva bitti. Tämä voi vaatia esimerkiksi saman kellopulssin käyttämisen molemmissa laitteissa. Synkroninen kommunikointi on useimmiten tehokkaampaa, sillä osapuolten välillä lähetellään vain databittejä (Durda, F 1996).

Asynkronisessa kommunikoinnissa datapaketin alkuun lisätään ns. starttibitti. Tämä bitti antaa vastaanottajalle tiedon siitä, että dataa on tulossa. Sen jälkeen yksittäiset bitit lähetetään alkaen vähiten merkitsevästä bitistä. Jokaisen bitin lähetykseen kuluu täsmälleen sama aika, ja noin puolessa välissä bitin lähetykseen kuluva ajasta vastaanottaja selvittää, onko bitti 1 vai 0. Kun kaikki bitit on vastaanotettu, tarkastetaan pariteettibitti (mikäli sitä on käytetty) ja odotetaan stop-bittiä. Mikäli tätä bittiä ei vastaanoteta, oletetaan, että on tapahtunut virhe. Uuden paketin lähetys voidaan aloittaa heti edellisen paketin stop-bitin lähetyksen jälkeen. (Durda, F 1996.)

## 5.2 Digitaalinen lämpötila-anturi DS18S20

Dallas Semiconductors Inc:n kehittämä DS18S20 (kuva 7) on digitaalinen lämpötila-anturi, joka mittaa lämpötilat 9-bittisenä. Anturi kommunikoi mikroprosessorin kanssa 1-Wire-väylän välityksellä, ja se vaatii ainoistaan yhden datalinjan maan (0 V) lisäksi. Anturin mittaustemperatuuri on  $-55\text{ °C} \dots +125\text{ °C}$ . Tarkkuus on  $\pm 0,5\text{ °C}$ , kun mitattava lämpötila on  $-10\text{ °C} \dots +85\text{ °C}$ . Anturi voi ottaa tarvitsemansa virran suoraan datalinjalta, jolloin ei tarvita erillistä virtalähdettä. Tässä tilanteessa anturin käyttämää virtaa kutsutaan "loisvirraksi". Loisvirtaa käytettäessä virta syötetään ylös- ja alaspäin virtauksen kautta datalinjalta silloin, kun väylä on ylätilassa. Anturin sisäinen kondensaattori ( $C_{PP}$ ) varautuu ja tarjoaa anturin toiminnalle tarvittavan virran aina väylän ollessa "alhaalla". Anturin virta voidaan toki syöttää myös erillisen jännitelähteen kautta.

Jokaisella DS18S20-anturilla on oma yksilöllinen 64-bittinen sarjanumero, joten samalla 1-Wire-väylällä voi olla useita samanlaisia antureita. Kun tämä voidaan ottaa ohjelmallisesti huomioon, on näppärää käyttää yhtä mikrokontrolleria ohjaamaan useita antureita laajallakin säteellä.



KUVA 7. TO92-pakkauksessa oleva DS18S20-anturi (Vitronia 2012)

### 5.2.1 Sähköiset ominaisuudet ja pinnimäärytykset

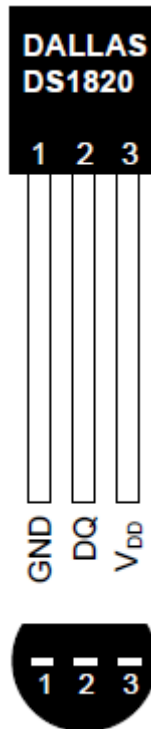
Anturin sähköisten ominaisuuksien tunteminen on erittäin tärkeää niin elektroniikan suunnittelun kuin ohjelmoinninkin kannalta. Taulukkoon 2 on kerätty tärkeimpiä sähköisiä ominaisuuksia.

TAULUKKO 2. DS18S20-anturin sähköisiä ominaisuuksia (Maxim Integrated Products 2010, 20)

Parametri	Symboli	Olosuhde	Minimi	Tyypillinen	Maksimi	Yksikkö
Syöttöjännite	$V_{DD}$	lokaali virta	+3.0		+5.5	V
Ylösvedon jännite	$V_{PU}$	loisvirta lokaali virta	+3.0 +3.0		+5.5 $V_{DD}$	V
Anturin virhe	$t_{ERR}$	-10°C...+85°C -55°C...+125°C			±0.5 ±2	°C
Nieluvirta	$I_L$	$V_{I/O} = 0.4V$	4.0			mA
Valmiusvirta	$I_{DDs}$			750	1000	nA
Mittausvirta	$I_{DD}$	$V_{DD} = 5.0V$		1	1.5	mA
D-linjan ottovirta	$I_{DQ}$			5		μA
Lämpötilan heittäly				±0.2		°C

Elektroniikan suunnittelun kannalta olennaisia anturin sähköisiä ominaisuuksia ovat syöttöjännite ja mittausvirta. Nämä suureet täytyy ottaa huomioon datalinjaa suunniteltaessa. Anturin on saatava riittävän suuri ulkoinen käyttöjännite ( $V_{DD}$ -pinniä käytettäessä) toiminnan varmistamiseksi. Liian suuri määrä antureita yhdellä linjalla taas voi mittauksista tehtäessä aiheuttaa väylän ylikuormittumisen. Tutustumalla huolellisesti taulukossa esitettyihin suureisiin ja suunnitteleamalla elektroniikka niiden mukaan,

vältytään laitteiden vikaantumisilta. Ohjelmoinnissa tulee huomioida mm. se, käyttääkö anturi loisvirtaa vai ulkoista jännitelähdettä. Näin anturille osataan lähettää oikeita komentoja, kun väylän liikennöintiin liittyvät ajastukset voidaan mitoittaa oikein. Tiedot viiveet mittaustoiminnoissa poikkeavat toisistaan, kun jännite syötetään anturille eri tavoin.



KUVIO 10. DS18S20 kuvattuna edestäpäin (Maxim Integrated Products 2010, 1)

Kuten yllä olevasta kuviosta 10 nähdään, pinni 1 kytketään maihin, pinni 2 on datalinjaa varten ja pinni 3 kytketään käyttöjännite. Mikäli anturin käyttöjännitteenä kuitenkin käytetään loisvirtaa, tulee pinni 3 kytkeä maihin, jotta voidaan varmistaa anturin oikea toiminta.

### 5.2.2 DS18S20-anturin protokolla

Tiedonsiirto DS18S20-anturin ja mikrotietokoneen välillä tapahtuu seuraavasti:

1. isäntä antaa nollauspulssin
2. isäntä kirjoittaa ROM-toimintakoodin
3. isäntä antaa muisti- tai ohjauskomennon
4. toiminta käynnistyy tai tietoa siirtyy

Nollauspulssin jälkeen isäntä antaa aina ROM-toimintakoodin (*ROM-function command*). ROM-toimintakoodit ovat 8-bittisiä koodeja, jotka voivat olla jokin seuraavasta viidestä koodista.

- **Read ROM (33H)** -komennolla luetaan yhden anturin 64-bittinen sarjanumero. Tätä komentoa voidaan käyttää ainoastaan silloin, kun väylälle on liitetty vain yksi anturi.
- **Match ROM (55H)** -komennolla osoitetaan tietyn numeroista anturia väylällä, johon on kytketty useita antureita. Tämän komennon perään annetaan 64-bittinen sarjanumero. Vain se anturi, jonka sarjanumero vastaa annettua numeroa, jää kuuntelulle. Muut anturit jäävät odottamaan seuraavaa nollauspulssia.
- **Skip ROM (CCH)** -komentoa käytetään säästämään aikaa väylällä, jonne on kytketty vain yksi anturi. Tämän komennon perään voidaan antaa suoraan muisti- tai ohjauskomento, ilman anturin 64-bittistä sarjanumeroa.
- **Search ROM (F0H)** -komennolla isäntä voi selvittää kaikkien väylää kytkettyjen antureiden sarjanumerot. Tämän komennon perään tulee melko mutkikas kirjoitus- ja lukualgoritmi, jonka avulla sarjanumerot saadaan selville.
- **Alarm Search (ECH)** -komennolla voidaan selvittää, onko jokin anturi hälytystilassa. DS18S20 anturissa on kaksi muistipaikkaa EEPROM-muistia, jonne voidaan tallentaa lämpötilan ala- ja ylärajat. Jos anturin lämpötila on näiden rajojen ulkopuolella, anturi on hälytystilassa.

(Koskinen 2004, 259.)

Kun ROM-toimintakoodi ja siihen liittyvä mahdollinen anturin numero on annettu, seuraavaksi annetaan muisti- tai ohjauskomento. Nämä komennot ovat myös 8-bittisiä koodeja.

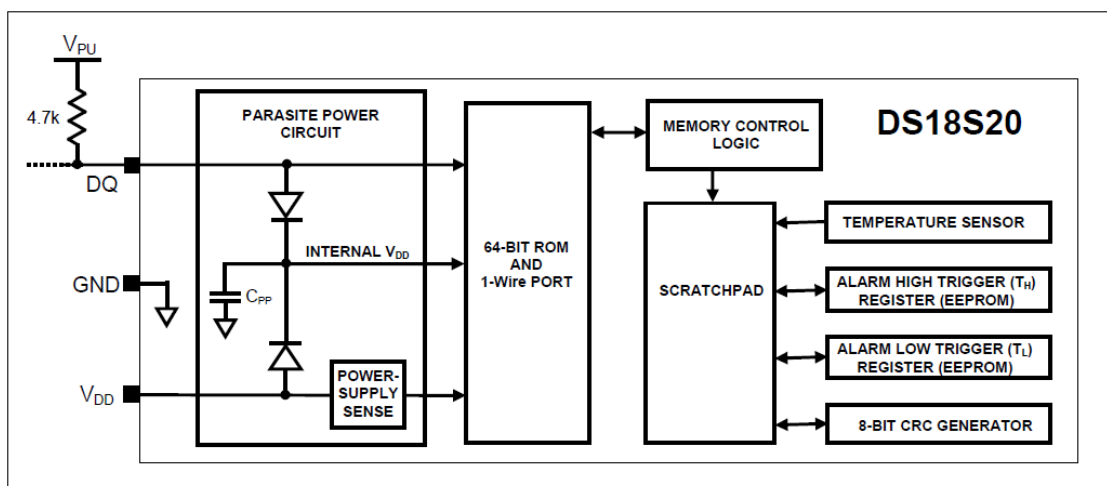
- **Write Scratchpad (4EH)** -komennolla voidaan kirjoittaa tietoa anturin RAM-muistiin. Ensimmäinen tieto kirjoitetaan muistipaikkaan kaksi ja seuraavat tiedot menevät aina seuraaviin muistipaikkoihin. Muistipaikat kaksi ja kolme ovat lämpötilan hälytysrajat (ylä- ja alaraja).
- **Read Scratchpad (BEH)** -komennolla voidaan lukea tietoa anturin RAM-muistista. Ensimmäinen tieto luetaan anturin ensimmäisestä muistipaikasta (osoite 0). Muistipaikasta 0 saadaan anturin lämpötilan kahdeksan alinta bittiä. Seuraavassa muistipaikassa (1) on lämpötilan yhdeksäs bitti. Viimeinen eli yhdeksäs tieto on tarkistussumma.
- **Copy Scratchpad (48H)** -komennolla voidaan anturin hälytysrajat kopioida muistipaikoista kaksi ja kolme anturin EEPROM-muistipaikkoihin.

- **Convert T (44H)** -komento käynnistää anturin lämpötilan mittauksen. Jos isäntä yrittää lukea anturia, jonka mittaus on kesken, saadaan tulokseksi aina nollabitti. Kun mittaus on suoritettu, anturin luku palauttaa ykköstilän.
- **Recall E<sup>2</sup> (B8H)** -komennolla voidaan kopioida anturin EEPROM-muistiin tallennetut hälytysrajat anturin RAM-muistiin. Kopiointi tapahtuu automaattisesti myös silloin, kun anturin käyttöjännite kytkeytyy.
- **Read Power Supply (B4H)** -komennolla voidaan anturista lukea, onko sille kytketty kiinteästi käyttöjännite vai ottaako anturi käyttöjännitteensä väylän data-johtimesta.

(Koskinen 2004, 259 – 260.)

### 5.2.3 Anturin toiminta

DS18S20-anturin toiminta on kuvattu lohkokaaaviona kuviossa 11. Lämpötilanmittaus toteutetaan käyttäen sensorilohkoa, jossa on kaksi jännitelähdettä, joista toisen jännitteen arvo vaihtelee lämpötilan vaikutuksen seurauksena, kun taas toinen pysyy vakiona. AD-muunnin muuntaa näiden jännitteiden erotuksen digitaalseksi arvoksi, joka esittää lämpötilaa. (Maxim Integrated Products 2009.)



KUVIO 11. DS18S20 lohkokaaaviona (Maxim Integrated Products 2010, 3)

DS18S20:n ulostulo on resoluutioltaan 9-bittinen. Mittauksen aloittamista ja A/D-muunnosta varten isännän (mikrokontrollerin) täytyy suorittaa komento Convert T [44h]. Muunnoksesta saatu lämpötilatieto tallennetaan 2-tavuiseen lämpötilarekisteriin, "scratchpad"-nimiseen muistiin. Tämän jälkeen anturi palaa "joutilaaksi". Mikäli anturin käyttöjännite tulee erillisestä jännitelähteestä, isäntä voi pyytää tätä lähettämään väylää pitkin merkin 0, jos lämpötilamuunnos on vielä kesken. Jos muunnos on

valmis, lähetetään 1. Mikäli käytetään loisvirtaa, tätä tekniikka ei voida käyttää. Tälöin väylä vedetään ylätilaan ylösvetovastuksen avulla koko muunnoksen ajaksi. (Maxim Integrated Products 2010, 3.)

DS18S20:n ulostulodata on kalibroitu Celsius-asteiksi. Mikäli halutaan käyttää Fahrenheit-asteikkoa, täytyy suorittaa erillinen muunnos tietyllä algoritmilla. Lämpötilatieto talletetaan 16-bittisenä etumerkillisenä kahden komplementtilukuna lämpötilarekisteriin (kuvio 12). Sign-bitit (S) kertovat sen, onko lämpötila negatiivinen vai positiivinen. Positiiviselle luvulle  $S=0$  ja negatiiviselle  $S=1$ . Tästä esimerkkinä taulukossa 3 on joitakin digitaalisia mittaustietoja ja niitä vastaavat lämpötilat. (Maxim Integrated Products 2010, 3.)

LS TAVU	BITTI7	BITTI6	BITTI5	BITTI4	BITTI3	BITTI2	BITTI1	BITTI0
	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$
MS TAVU	BITTI15	BITTI14	BITTI13	BITTI12	BITTI11	BITTI10	BITTI9	BITTI8
	S	S	S	S	S	S	S	S
S=SIGN								

KUVIO 12. DS18S20:n lämpötilarekisteri

TAULUKKO 3. Eri lämpötilojen ja binääristen ulostulojen vastaavuudet (Maxim Integrated Products 2010, 4)

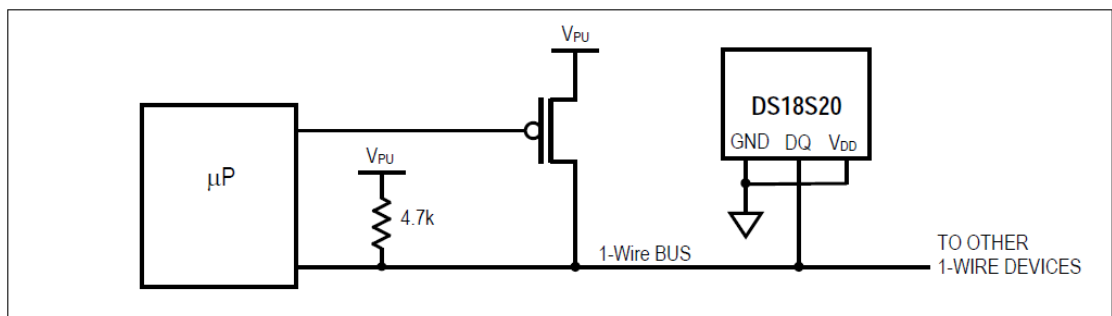
LÄMPÖTILA (C°)	DIGITAALINEN ULOSTULO (BIN)	DIGITAALINEN ULOSTULO (HEX)
+85.0	0000 0000 1010 1010	00AAh
+25.0	0000 0000 0011 0010	0032h
+0.5	0000 0000 0000 0001	0001h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1111	FFFFh
-25.0	1111 1111 1100 1110	FFCEh
-55.0	1111 1111 1001 0010	FF92h

#### 5.2.4 Virransyöttö

DS18S20:n tarvitsema käyttöjännite voidaan syöttää joko  $V_{DD}$ -pinniin (kuvio 14), tai vaihtoehtoisesti käyttää loisvirtaa (kuvio 13), jolloin virta otetaan Datalinjalta 1-Wire-väylän ollessa ylätilassa. Tällöin myös anturin sisäinen kondensaattori  $C_{PP}$  varautuu ja kykenee pitämään anturin toiminnassa myös silloin, kun väylä on alatilassa. Loisvirtaa käytettäessä 1-Wire-väylä ja  $C_{PP}$  pystyvät tarjoamaan riittävän virran DS18S20:lle suurimmassa osassa toimintoja niin kauan kun ajoituksiin ja jännitteisiin

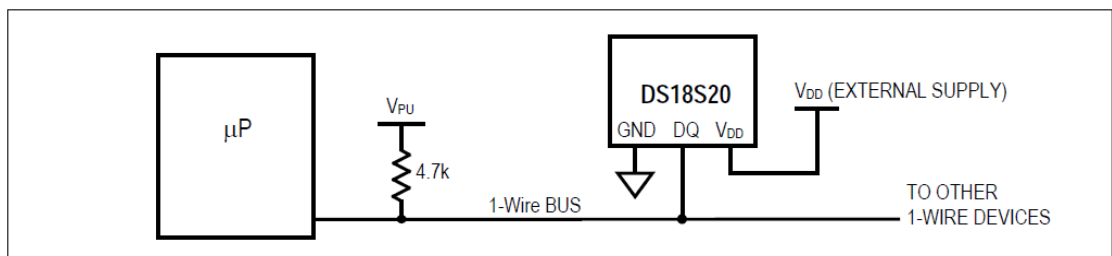


liittyviin vaatimuksiin vastataan. Kuitenkin mikäli suoritetaan lämpötilan muunnosta tai kopioidaan dataa scratchpadista EEPROMille, virta voi olla jopa 1.5 mA. Tämä virta voi aiheuttaa ei-toivotun jännitteen heittelyn. Jotta voitaisiin varmistaa anturin riittävä virta, täytyy 1-Wire-linjalle lisätä voimakas ylösveto edellä mainittuja toimintoja varten. Tämä saadaan aikaiseksi käyttämällä MOSFETia vetämään väylä oikeaan aikaan ylös. Väylä on kytkettävä ylösvetoon 10  $\mu\text{s}$ :n sisällä (Max.) Convert T [44h] tai Copy Scratchpad [48h] -käskyn jälkeen. Väylän täytyy pysyä ylätilassa muunnoksen tai tiedonsiirron ajan. Tällä aikaa väylällä ei saa olla muita tapahtumia. (Maxim Integrated Products 2010, 5.)



KUVIO 13. Anturin ja mikroprosessorin välinen kytkentä käytettäessä lois-virtaa (Maxim Integrated Products 2010, 6)

DS18S20:n virta voidaan syöttää myös perinteisellä tavalla kytkemällä  $V_{DD}$ -pinniin erillinen jännitelähde. Etuina ovat mm. se, ettei MOSFETia tarvita sekä se, että esim. muunnoksen aikana väylällä voi olla liikennettä. Lisäksi, mikäli mitataan yli 100 °C:n lämpötiloja, on suositeltavaa käyttää erillistä jännitelähdettä. Tämä siksi, että näin korkeissa lämpötiloissa voi esiintyä virtavuotoja, jotka aiheuttavat vääristymiä mittauksiin.



KUVIO 14. Anturin ja mikroprosessorin välinen kytkentä käytettäessä perinteistä ulkoista jännitelähdettä (Maxim Integrated Products 2010, 6)

### 5.2.5 64-bittinen ROM-koodi

Kaikilla DS18S20-antureilla on uniikki 64-bittinen ROMille tallennettu "sarjanumero" (kuvio 15). Vähiten merkitsevät kahdeksan bittiä sisältävät DS18S20:n tuoteperhe-numeron, joka on 10h. Seuraavat 48 bittiä käsittävät yksilöllisen sarjanumeron, jollaista ei voi olla millään muulla anturilla. Kahdeksan eniten merkitsevää bittiä ovat eräänlaista virheentarkistuskoodia (cyclic redundancy check, CRC) varten. Tämä tavu lasketaan ROM-koodin ensimmäisestä 56 bitistä. Kokonaisuudessaan 64-bittinen ROM-koodi mahdollistaa DS18S20:n noudattaa tiettyä protokollaa ja toimia osana 1-Wire-väylää. (Maxim Integrated Products 2010, 6.)

8-BIT CRC		48-BIT SERIAL NUMBER				8-BIT FAMILY CODE (10h)	
MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB

KUVIO 15. DS18S20:n 64-bittinen ROM-koodi (Maxim Integrated Products 2010, 6)

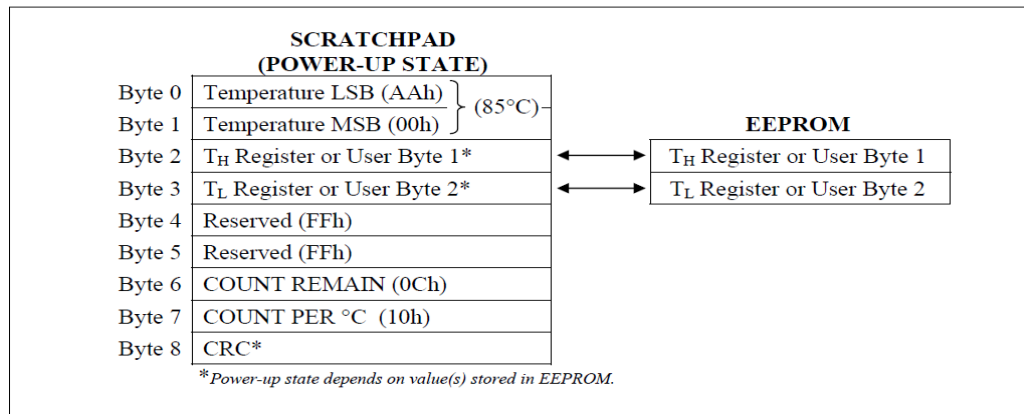
### 5.2.6 Muisti

Anturin muisti koostuu SRAM- sekä EEPROM-muisteista kuvion 16 mukaisesti. SRAM on lämpötiladataa varten ja haihtumaton EEPROM mahdollisia ohjelmoitavia lämpötilan ylä- ja alahälytysrajoja varten. Niitä varten on rekisterit  $T_H$  ja  $T_L$ . Mikäli näitä rekistereitä ei käytetä hälytysrajojen ohjelmoimiseen, ne palvelevat normaalina muistina (Maxim Integrated Products 2010, 7).

Scratchpad-alueen tavut 0 ja 1 sisältävät vähiten ja eniten merkitsevät bitit lämpötilarekisteristä, ja ne ovat read-only – tyyppiä. Tavut 2 ja 3 ovat  $T_H$ - ja  $T_L$  – rekistereitä varten. Tavut 4 ja 5 ovat varattuina laitteen sisäiseen käyttöön, eikä niihin voi kirjoittaa mitään. Tavuissa 6 ja 7 on COUNT REMAIN- ja COUNT PER °C-rekisterit, joita voidaan käyttää laskettaessa laajemman resoluution tuloksia. Tavu 8 sisältää CRC-koodin. (Maxim Integrated Products 2010, 7.)

Data kirjoitetaan scratchpadin tavuihin 2 ja 3 käyttämällä komentoa Write Scratchpad [4Eh]. Tiedonsiirto DS18S20:lle aloitetaan tavun 2 vähiten merkitsevästä bitistä. Dataneheys voidaan varmistaa lukemalla scratchpadin tieto komennolla Read Scratchpad [BEh] kirjoituksen jälkeen. Luettaessa scratchpadia, data siirretään 1-Wire-väylän yli aloittaen tavun 0 vähiten merkitsevällä bitillä. Kun  $T_H$ :n ja  $T_L$ :n dataa siirretään EEPROMille, isännän täytyy antaa komento Copy Scratchpad [48h]. EEPROMin data säilytetään, vaikka laitteen virta olisi poissa. Kun laitteeseen kytketään taas virta, data

ladataan vastaaviin scratchpadin muistipaikkoihin. Data voidaan myös ladata EEPROMilta scratchpadiin milloin tahansa käyttämällä Recall E<sup>2</sup> [B8h] – komentoa. (Maxim Integrated Products 2010, 7.)



KUVIO 16. DS18S20:n muistikartta (Maxim Integrated Products 2010, 7)

### 5.2.7 CRC-koodi

Tarkistussumma on tietotekniikassa käytetty tarkistuskoodaustapa, jolla voidaan todeta, onko siirretty tai tallennusmedialta luettu tieto ehjää vai siirto- tai tallennusvirheiden sotkema. Tarkistussumma toimii siten, että datasta muodostetaan tiivistealgoritmillä vakiokokoinen luku. Datan oikeellisuus voidaan tarkistaa laskemalla tarkistussumma uudelleen. Jos summa ei täsmää edellisen summan kanssa, data on virheellistä. Yksinkertaisen tarkistussumman eräs heikkous on se, että toisaalla datassa oleva virhe voi varsin helposti kompensoida toisen virheen ja näin kaksi toisensa kumoavaa pientä virhettä saa datapaketin näyttämään ehjältä. Tarkistussummaa tehokkaampi koodi tarkistukseen on esimerkiksi CRC (cyclic redundancy check) jolla voidaan välttää edellä kuvattu parin pienen virheen sisältävän datan tulkitseminen ehjäksi kokonaisuudeksi. (Wikipedia 2011b.)

CRC on osa DS18S20:n 64-bittistä ROM-koodia. ROM-koodin CRC lasketaan ROMin ensimmäisestä 56 bitistä ja sitä säilytetään ROMin eniten merkitsevässä tavussa. Scratchpadin CRC taas lasketaan scratchpadiin varastoidusta datasta ja se näin ollen vaihtuu datan vaihtuessa. CRC auttaa isäntäohjainta varmentamaan datan oikeellisuuden luettaessa dataa DS18S20:lta. Jotta datan luku voitaisiin todeta oikeanlaiseksi, isännän täytyy laskea CRC uudelleen ja verrata saatua arvoa joko ROMin CRC:hen tai scratchpadin CRC:hen. Mikäli arvot täsmäyvät, data on vastaanotettu virheettömänä. (Maxim Integrated Products 2010, 8.)

## 6 ANTURIVERKKO

Koska mittauspisteet sijaitsivat melko laajalla alueella eri huoneissa hajautetusti, päätettiin tehdä kaksi pienempää erillistä verkkoa. Näin saatiin jaettua lämpimät ja kylmät mittauspisteet omiksi kokonaisuuksiksi. Nimitetään näitä kahta eri verkkoa kirjaimilla A ja B. Mittauslaitteistolta alueen kahdeksi jakaminen ei vaatinut kuin yhden Arduino Duemilanoven lisää. PC:ssä riittää USB-portteja, joihin saadaan liitettyä kaksi mittalaitetta. Kaksi erillistä verkkoa helpottaa asennustyötä sekä vikatilanteissa virheen paikannusta.

Anturiverkon kaapeloinnin toteutukseen on olemassa monia tapoja. Toteutusta suunniteltaessa tuli ottaa huomioon mm. käytännöllisyys ja siisteys. Kaupallisuus ja käytännöllisyys huomioiden tarkoitus oli löytää kaapelimalli, jonka avulla verkko olisi helppo koota ja haaroittaa mittauspisteisiin ”plug and play”-idealla. Valmistajan omassa suunnitteluoppaassa yhtenä yleisenä kaapelointivaihtoehtona esiteltiin Cat5-tyypin neljän parin kaapeli. Kaapelista kerrotaan tarkemmin seuraavaksi. Esitelty kaapeli sopii hyvin 1-Wire-laitteille, sillä saman muovikuoren alla saadaan kuljetettua datasiinaalit, käyttöjännite antureille sekä referenssimaa.

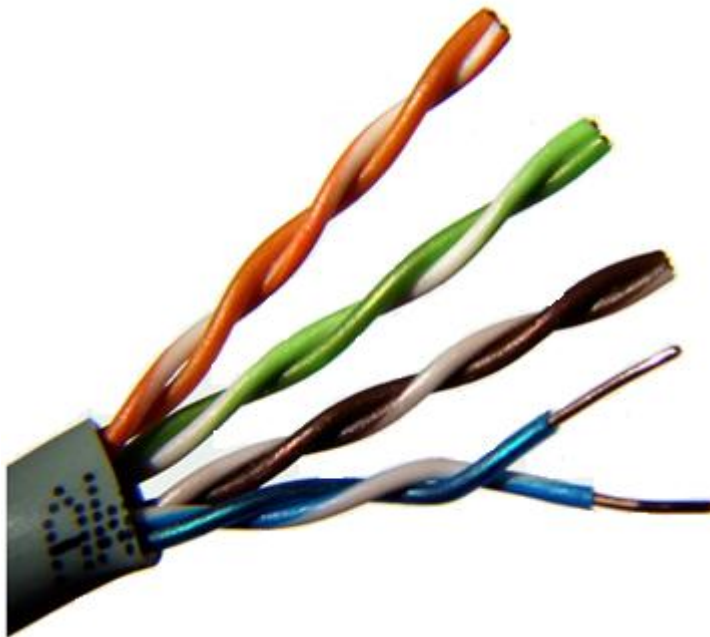
### 6.1 CAT5 UTP -kaapeli

Anturiverkon onnistumisen kannalta oikeanlainen kaapeli on aina avainasemassa. Koska 1-Wire-väylän vähimmäismäärä johtimille on kaksi, riittäisi kaapeloinniksi periaatteessa vaikka normaali parikaapeli. Tämä ei kuitenkaan ole paras mahdollinen ratkaisu etenkin, mikäli kokonaisuudesta haluaa häiriönsuojaukseltaan ja siisteydeltään parhaan lopputuloksen. Aihetta käsittelevien datalehtien ja artikkeleiden perusteella käytetyin ja suositelluin kaapeli 1-Wire-verkkoon on Cat5 UTP-tyypin ethernet-kaapeli. Kaapelin todettiin olevan sopiva myös tulevaan ympäristöön mm. yleisesti käytettyjen liittimiensä vuoksi.

Parikaapeli on yleinen kaapelityyppi, jossa käytetään toistensa ympäri kierrettyjä johdinpareja häiriöiden vähentämiseksi. Parien kierto on tyypillisesti kolme kierrosta tuumalla. Jokaisella parilla on erisuuruinen parikierto häiriöiden parista toiseen siirtymisen estämiseksi. Parikaapeleita voidaan käyttää muun muassa Token Ring -, puhelin-, Ethernet-, FDDI-, ISDN- ja ATM-yhteyksiin. Yleisin liitintyyppi parikaapeleille lähiverkossa on RJ-45. Kaapeli ja parit voidaan ympäröidä metallivaipalla, jolloin saa-

vutetaan paras suoja häiriöitä vastaan. Tällaista parisuojattua STP-kaapelia (*Shielded Twisted Pair*) käytetään silloin, kun suojaukseen on erityistä tarvetta tai kaapelin luokka vaatii sen (esimerkiksi Cat7). Suojaustarve voi syntyä monesta syystä, kuten muuntajista, voimavirtakaapeleista, oikosulkumoottoreista tai muista voimakkaita magneettikenttiä aiheuttavista seikoista, staattisen sähköön purkauksista, kipinöinnistä, valokaarista jne. Tavallisin kaapelityyppi on kuitenkin suojaamaton UTP (*Unshielded Twisted Pair*), jota käytetään puhelinverkoissa ja tietoliikennetekniikassa. Lisäksi on olemassa suojattu FTP-kaapeli (*Foiled Twisted Pair*), joka on periaatteessa STP ilman parikohtaista suojausta. Kierretyt parikaapelit on jaettu eri kategorioihin niiden kaistanleveyksien mukaan. (Wikipedia 2009.)

Tässä työssä riittävä valinta oli CAT5 UTP -kaapeli (kuva 8). Laboratorion lämpökaapit toimivat normaalilla 230 V:n verkkojännitteellä, eikä lähettyvillä ole suuria voimavirtalaitteita, joten suuria häiriöitä mittaushetkillä ei ilmene.



KUVA 8. CAT5 UTP -kaapeli (Made-in-china.com 2012)

## 6.2 Haaroitus

Antureiden kytkeminen verkkoon tuli olla helppoa, joten täytyi löytää keino, jolla kaapeli saatiin haaroitettua aina mittauspisteen kohdalla ja kytkettyä anturi haaraan ongelmitta. Koska anturit tulee kytkeä rinnakkain, oli haaroittimen oltava rinnankytkävä RJ45-standardin 1in 2out-liitin. Vaatimuksiin sopivia malleja oli melko vähän saatavil-

la, ja niiden hinta oli 3 € - 15 €. Edullisin sopiva haaroitin oli myytävänä Bebekin verkko-  
kokaupassa. Kuva 9 esittää työssä käytettyä haaroitinta.



KUVA 9. Haaroittimena käytetty LAN MODULAR 08 PIN (Bebek Electronic 2012)

### 6.3 Verkon terminologiaa

1-Wire-verkkoon liittyy tiettyjä suureita, jotka vaikuttavat osaltaan sen toimintaan. Kaksi tärkeää termiä ovat säde ja paino. Verkon säde kuvaa etäisyyttä isännän ja etäisimmän orjan välillä. Säde mitataan metreinä. Verkon paino tarkoittaa käytetyn kaapelin kokonaismäärää. Sekin mitataan metreinä. Paino rajoittaa signaalin nousu-aikaa kaapelissa, kun taas säde vaikuttaa hitaimman (signaalin) heijastuksen ajoitukseen. Myös jokainen orja lisää verkon painoa. Koska verkon paino voi olla rajallinen, voidaan sanoa, että mikäli orjia on paljon, voidaan kaapelin määrää joutua vähentämään.

Perinteinen ylös- ja alaspäin suuntautuvalla toteutettu linja kykenee ainakin 200 m:n painoon, joten tämän projektin verkossa ongelmia painon takia ei missään vaiheessa ilmennyt. Tämä voidaan osoittaa myös laskemalla toteutettujen verkkojen painot seuraavien tiedoin:

- Jokainen 1-Wire-laite lisää painoa 0,5 m.
- Kaapelia on yhteensä n. 25 m verkkoa kohden.
- Orjia on verkossa A 8 kpl ja verkossa B 5 kpl.

Verkon A paino:

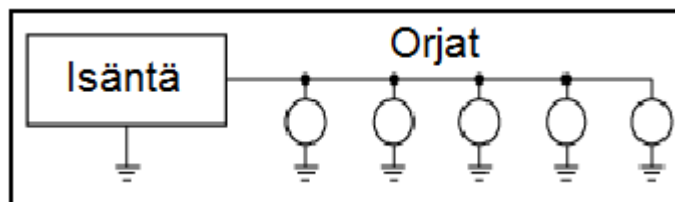
$$25\text{ m} + 8 * 0,5\text{ m} = 29\text{ m}$$

Verkon B paino:

$$25\text{ m} + 5 * 0,5\text{ m} = 27,5\text{ m}$$

#### 6.4 Lineaarinen verkkotopologia

Tärkeä huomioon otettava seikka missä tahansa verkossa on verkkotopologia. Erilaisista topologioista voidaan mainita lineaarinen -, tähti - ja pätkätopologia. Näistä suositteluin on lineaarinen topologia (kuvio 17). Siinä verkko alkaa isännästä ja jatkuu aina kauimpana olevaan orjaan saakka.



KUVIO 17. Lineaarinen verkkotopologia (Maxim Integrated Products 2008)

Orjat kytketään datalinjalle lyhyin (< 3 m) haaroituksin. Tämä topologia on juuri sopiva rakennettuihin mittausverkkoihin, sillä linja voidaan vetää mikrokontrollerilta aina viimeiselle mittauspisteelle asti lisäämällä haarakappale (anturi + verkon haaroitin) aina kunkin mittauspisteen kohdalle (Maxim Integrated Products 2008).

## 7 PC:N JA MIKROKONTROLLERIN VÄLINEN RAJAPINTA

Olennainen osa laitteistoa on mikrokontrollerin sekä PC-tietokoneen toimiva keskinäinen kommunikointi. Fyysinen yhteys toimii USB-kaapelin välityksellä. Arduino Duemilanoven eräänä ominaisuutena on RS-232-USB-konvertteripiiri, joka muuntaa sarjaliikennetasoiset signaalit sopivaksi USB-liitännälle. Tämä on mielestäni hyvä ominaisuus, sillä USB-liitäntä on näppärämpi mm. kokonsa sekä helpomman liitäntänsä takia. Lisäksi nyky-PC-tietokoneissa on harvoin RS-232-portteja.

### 7.1 RS-232

RS-232 (*Recommended Standard 232*) on kahden tietokonelaitteen väliseen tietoliikenteeseen tarkoitettu portti, jossa data siirtyy yksi bitti kerrallaan ”peräkkäin” asynkronisesti sarjamuotoisena. Portista käytetään yleisesti nimitystä sarjaportti tai com-portti. Kommunikoivat laitteet voivat olla esimerkiksi pääte (DTE = Data Terminal Equipment) ja kommunikaatiolaite (DCE = Data Communication Equipment). RS-232-signaalit ovat tasoiltaan noin  $\pm 5$  V.. $\pm 12$  V suhteessa maatasoon. RS-232-porttia käyttävät laitteet päihittävät yksinkertaisuudessaan muita portteja käyttävät laitteet, minkä vuoksi sitä onkin käytetty PC-koneissa pisimpään. (Wikipedia 2011a).

### 7.2 Tiedonsiirto laitteiden välillä

Jotta yhteys saadaan toimimaan halutulla tavalla, tulee mikrokontrollerissa sekä PC:ssä olla ohjelmoituna oikeat funktiot, joissa merkkien lähettely laitteiden välillä määritellään. Nämä funktiot ovat osa Windowsin suurta API-rajapintaa (API = Application Programming Interface) (Maksimainen, V 2009).

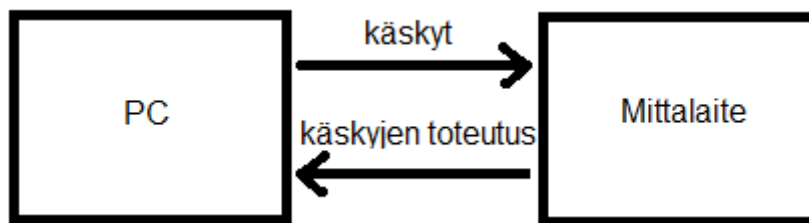
Luetellaan sarjaportin käyttöön liittyvät vaiheet:

- Sarjaportin avaus. Windowsissa käytetään tähän tarkoitukseen sen omaa funktiota CreateFile. Tässä määritellään mm. käytettävän portin numero, portin käyttö kirjoitukseen ja lukemiseen, synkroninen/asynkroninen käyttö (tässä käytetään synkronista).
- Asetetaan liikennöintiparametrit samoiksi kuin mittalaitteella. Näitä parametreja ovat mm. liikennöintinopeus (baudrate) ja pariteettibitin käyttö.



- Tämän jälkeen sarjaliikenne on käytettävissä. Mittalaitteelle menevät merkit lähetetään WriteFile-funktiolla ja mittalaitteelta tulevat merkit luetaan ReadFile-funktiolla.
- Lopuksi sarjaportti vapautetaan CloseHandle-funktiolla.

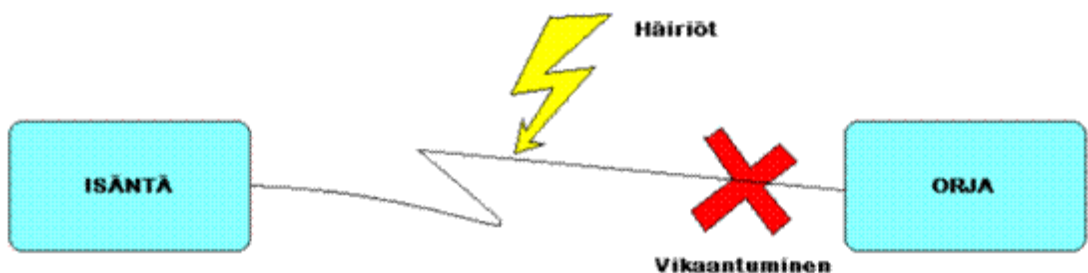
Tiedonsiirto toteutetaan siten, että PC toimii isäntänä ja mittalaite orjana. Kaikki tietoliikenne tapahtuu isännän aloitteesta ja orja toteuttaa isännän antamat tehtävät, kuten kuvio 18 nähdään.



KUVIO 18. Tiedonsiirrossa isäntä kääsee ja orja vastaa

### 7.3 Luotettavan tiedonsiirron toteuttaminen

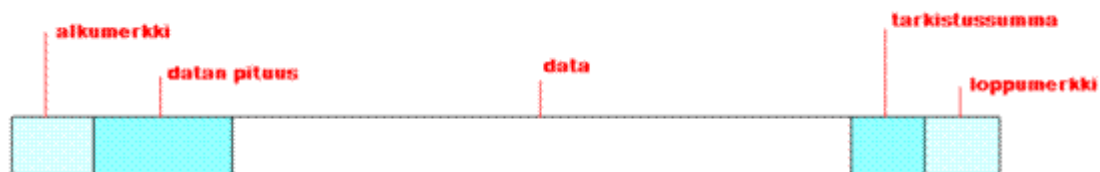
Laitteiden välinen tietoliikenne on hyvin häiriöaltis (kuvio 19). Sarjatieliikenne on toteutettu yhdistämällä kommunikoivat laitteen johtimilla, joista osa kuljettaa dataa ja osa on nk. kontrollisignaaleja. Kaikki johtimet vastaanottavat radiotaajuisia häiriöitä, jotka muuttavat alkuperäistä signaalia. Lähetty merkki saattaa muuttua toiseksi merkiksi tai kadota kokonaan. Jopa ylimääräisten merkkien syntyminen on mahdollista (Savonia 2009).



KUVIO 19. Tiedonsiirrossa laitteiden välillä voi esiintyä häiriöitä (Maksimainen, V 2009)

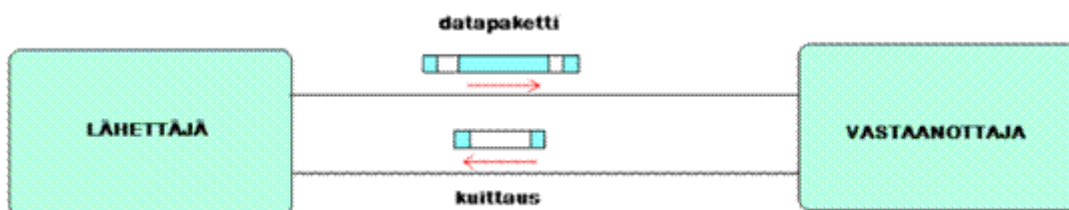
Jotta tietoliikenne saadaan toimimaan luotettavasti häiriöllisessä ympäristössä ja että se toipuu vikatilanteiden jälkeen, tiedonsiirtoon tarvitaan tarkkaan määritelty toimintatavat eli protokolla. Protokollan perusidea on lähettää viestit paketteina (kuvio 20), joilla on tarkkaan määritelty muoto. Tämän työn laitteissa käytetyssä protokollassa

paketti alkaa alkumerkillä ja päättyy loppumerkkiin. Alkumerkin perässä lähetetään datan pituus ja sitten itse data. Tarkistussummassa on laskettu yhteen datatavut ja datan pituus. Vastaanottopäässä suoritetaan samat laskut ja varmistetaan, että saatiin sama summa. (Savonia 2009.)



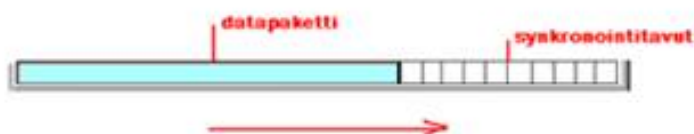
KUVIO 20. Viestit lähetetään paketteina (Maksimainen, V 2009)

Vastaanotetusta paketista tulee löytyä myös oikea alku- ja loppumerkki. Jos paketista ei löydy huomauttamista, vastaanottaja lähettää kiittauksen saadusta paketista (kuvio 21). Viallinen paketti pyydetään lähettämään uudestaan. Lähetystä yritetään uusia kolme kertaa.



KUVIO 21. Datapaketti kuitataan vastaanotetuksi (Maksimainen, V 2009)

Menettely varmistaa, että vastaanotetussa datassa ei ole vikaa. Virhe on mahdollinen, mutta sen todennäköisyys on hyvin pieni, koska virheiden tulisi summautua dataan vastakkaismerkkisinä lukuarvoina ja olla summa-arvoltaan nolla. Jos dataosan pituus vaihtelee paljon eri pakettien kesken, on mahdollista että datan sisällä on dataa, jonka tavut ovat oikean datapaketin muotoisia. Tässä tapauksessa vastaanottaja voisi virheellisesti tulkita datan osaa erilliseksi datapaketiksi. Tämän vuoksi jokaisen datapaketin edessä lähetetään nk. synkronointitavuja (kuvio 22), joilla eliminoidaan paketin tulkinnan aloitus väärästä paikasta. (Maksimainen, V 2009.).



KUVIO 22. Datapaketti sisältää synkronointitavuja (Maksimainen, V 2009)

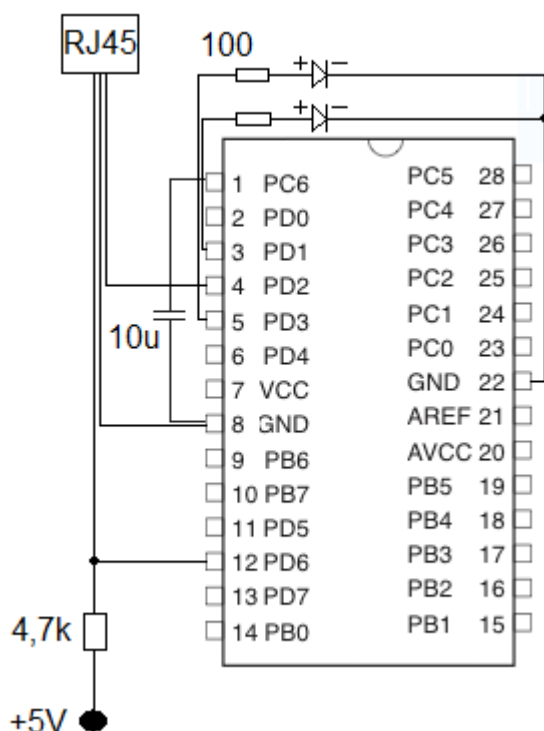
Tässä protokollassa siis käytetään moninkertaista varmistusta, joka parantaa tiedon siirron luotettavuutta.

## 8 LAITTEISTON RAKENTAMINEN

Mielenkiintoisin osa-alue koko työssä oli ehdottomasti laitteiston rakentaminen. Se piti sisällään ohjeiden ja vaatimusten mukaiset mikrokontrolleriin tehtävät kytkennät, mikrokontrollerin koteloinnin ja kotelon muotoilun, antureiden kaapeloinnin ja suoja-uksen sekä verkon rakentamisen.

### 8.1 Johdotukset ja kytkennät mikrokontrollerissa

Mikrokontrolleriin (Atmega328P) täytyi tehdä erilaisia kytkentöjä, jotta se saatiin toimimaan halutulla tavalla. Aiemmissa luvuissa esiteltiin mm. 1-Wire-väylän vaatimukset. Lisäksi kotelon pinnalle haluttiin LED-lamput indikoimaan laitteen toimintaa. ATmega328P:n pinneistä käytettiin lähinnä digitaalisia I/O-pinnejä, sillä niillä saatiin helpoiten toteutettua kaikki tarvittava. Kuviossa 23 esitetään mikrokontrolleriin tehdyt kytkennät piirikaaviona.

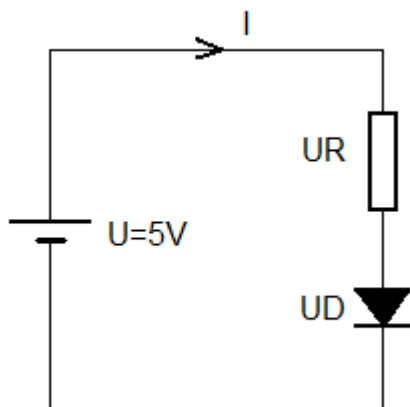


KUVIO 23. Atmega328P:hen tehdyt kytkennät piirikaaviona

Yllä olevassa kuvassa näkyy mikrokontrolleriin tehdyt kytkennät. Tarkastellaan ensin RJ45-porttiin menevät signaalit. 1-Wire-linja tarvitsee +5 V:n jännitteen, josta se kyt-

ketään ylösvetovastuksen kautta datalinjaan (pinni PD6). Referenssiksi linja tarvitsee maan (0 V). Lisäksi loisivirran sijaan päätettiin syöttää antureilla erillinen käyttöjännite (+ 5 V), joka saadaan ohjelmallisesti pinnistä PD2. Edellä mainitut ovat verkon CAT5-kaapelissa käytetyt signaalit. Työssä käytetyllä mikrokontrollerilla on ominaisuus, joka aiheuttaa resetoinnin joka kerta, kun sarjaliikenneportti vastaanottaa merkin. Tämä ei ole suotavaa, sillä koodin suorittaminen alkaisi alusta aina, kun jokin PC:n komento otettaisiin vastaan. Tähän ongelmaan löytyi kuitenkin ratkaisu. Reset- ja GND-pinnien väliin tuli kytkeä kondensaattori, joka tavallaan ”nielaisee” reset-pulssin. PD1- ja PD3 pinnejä käytetään LED-lamppujen ohjaamiseen.

Koska laitteeseen haluttiin myös kaksi eriväristä lediä, huomioon täytyy ottaa myös niiden vaatimukset ja kytkennät. 100 ohmin etuvastukset olivat siihen sopiva valinta, ja kun molemmat käytetyt lamput kytkettiin eri pinneihin, pysyi virrankulutus sallituissa rajoissa. Ennen etuvastuksen valintaa täytyi tehdä hieman laskutoimituksia. Asioita, jotka täytyy tietää, ovat ledin kynnyksjännite (2 V), syöttöjännite (pinnin 5 V) ja tarvittu virta (n. 30 mA). Kuviossa 24 on kytkentäkaavio LEDien etuvastuksen mitoittamiseen.



KUVIO 24. LEDin etuvastuksen mitoituksen kytkentäkaavio

Näillä tiedoilla voidaan laskea tarvittu etuvastuksen suuruus. Kyseessä on sarjaan kytkentä, joten virta on saman suuruinen vastuksessa ja ledissä. Jännite jakaantuu molemmille komponenteille, eli

$$U_R + U_D = 5 \text{ V.}$$

Tästä saadaan  $U_R$ :lle:

$$5 \text{ V} - 2 \text{ V} = 3 \text{ V.}$$

Resistanssi voidaan laskea ohmin lain avulla, jolloin

$$U_R / I = 100 \, \Omega.$$

Johdotukset tehtiin käyttämällä Arduinolle tarkoitettuja hyppylankoja, ja kaikkiin mahdollisiin liitoksiin tehtiin juotokset (kuva 10).



KUVA 10. Arduino Duemilanove ja siihen tehdyt kytkennät (Jesse Nissinen 2012)

## 8.2 Mikrokontrollerin kotelointi ja kotelon muotoilu

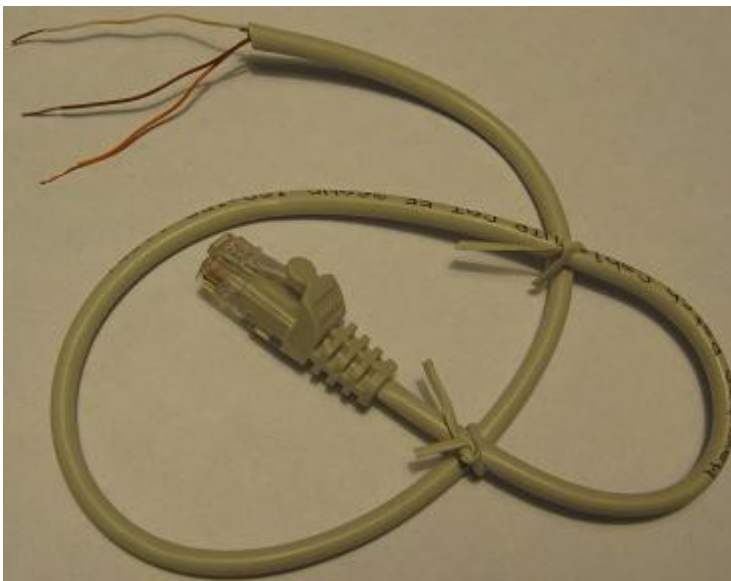
Kotelointi toteutettiin Arduinolle sopivalla mustalla muovikotelolla (kuva 11). Koteloon muotoiltiin tarpeelliset aukot led-lampuille sekä RJ45-liittimelle. USB-portille sekä virtaliitännälle aukot olivat jo valmiina.



KUVA 11. Valmis koteloitu mittalaite (Jesse Nissinen 2012)

### 8.3 Antureiden juottaminen ja suojaaminen

Koska anturit tuli saada haaroitettua pääkaapelista kaappien sisään, juotettiin jokainen anturi kiinni lyhyeen johdon pätkään. Jokainen lyhyt johto on 60 cm pitkä, joka on riittävä pituus.

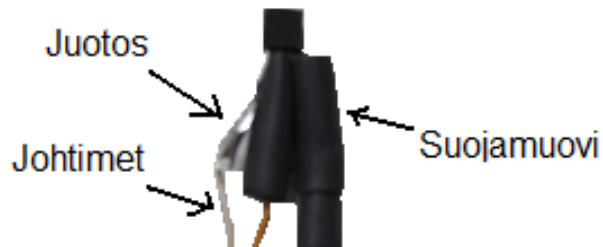


KUVA 12. Anturikaapeli (Jesse Nissinen 2012)

Anturin pinnit juotettiin erivärisiin johtimiin seuraavalla tavalla:

- oranssi johdin: datapinni (anturin keskimmäinen pinni)
- oranssi/valkoinen johdin: maapinni (anturin vasen pinni)
- ruskea johdin: +5 V  $V_{DD}$  -pinni (anturin oikea pinni)

Antureiden suojaukseen käytettiin itseliimautuvaa kuumennettavaa kutistemuovia. Näitä tarvittiin kahta halkaisijaltaan erikokoista.



KUVA 13. Anturin suojaaminen kutistemuovilla (Jesse Nissinen 2012)

Ensin kaapelin toinen rj45-liitin katkaistiin pois ja johtimet kuorittiin esiin (kuva 12). Halkaisijaltaan pienempi kutistemuovi pujotettiin johtimien ympärille ennen juotoksia. Juottamisen jälkeen kutistemuovit kutistettiin johtimien ja juotoksen päälle (kuva 13). Kutistaminen onnistui mainiosti lämmittämällä muovia hiustenkuivaajalla. Kuumetessa muovi surkastuu, ja siinä oleva liima sulaa kiinni suojattavaan kohteeseen. Lopuksi anturin ja suojattujen johtimien päälle laitettiin paksumpi kutistemuovin pätkä, jolloin lopputuloksena oli siisti ja hyvin suojattu anturikaapeli (kuva 14).



KUVA 14. Suojattu ja viimeistelty anturikaapeli (Jesse Nissinen 2012)



#### 8.4 Anturiverkon haaroitus ja kaapelointi

Verkon haaroitukseen löytyi sopivia aiemmin esiteltyjä rinnankytkeviä RJ45-haaroittimia. Haaroittimeen tulevat signaalit ovat samat sen molemmissa lähtevissä porteissa. Kuvista 15 ja 16 nähdään osa anturiverkon valmiista asennuksesta. Mikrokontrollerilta lähtevä kaapeli on mahdollista haaroittaa aina mittauspisteen kohdalla. Toiseen haaraan kytketään anturijohto ja toiseen linjaa jatkava johto. Asennettaessa haaroitettu anturikaapeli vietiin läpi kaappien ovien tiivisteiden välistä. Kaapelit saatiin kiinnitettyä siististi kaappien seinämiin peitelistoilla.



KUVA 15. Anturiverkon haaroitus lämpökaapeille (Kari Ojanen 2012)



KUVA 16. Anturiverkon kaapelointi lämpökaapeille (Kari Ojanen 2012)



## 9 KÄYTTÖLIITTYMÄN VAATIMUKSET

Järjestelmän haluttiin keräävän lämpötiladataa PC:lle. Data tuli arkistoida kovalevylle tekstitiedostoihin helposti tarkasteltavaksi. Sen täytyi myös olla analysoitavissa Excelissä. Seurantaväliksi haluttiin yksi tunti, joten vuorokaudessa mittauksia tulee 24 kappaletta. Mittaustulokset haluttiin päiväkohtaiseen tekstitiedostoon, jossa lukemat lajitellaan oikeille paikoilleen mittauspisteiden ja tuntien mukaan. Kullekin anturille haluttiin myös oma kuukausikohtainen Excel-tiedostonsa, jossa kuukauden jokaiselle päivälle on oma rivinsä taulukossa. Tähän taulukkoon täytyi kerätä ainoastaan päiväkohtaiset keskiarvot, minimi ja maksimit.

## 10 OHJELMOINTI

Koko työn suurin ja aikaa vievin osa oli ohjelmiston toteuttaminen. Vaatimuksena oli toteuttaa yksinkertainen ja kaikin puolin selkeä kokonaisuus ja käyttöliittymä, jota olisi kenen tahansa helppo käyttää. Koko järjestelmässä on kolme eri ohjelmaosiota: mittalaitteen ohjelma, PC:n tiedonkeruuohjelma sekä Excel-käyttöliittymä. Aiempaa kokemusta ohjelmoinnista oli muutamien kurssien verran, joiden aikana käytettiin keskeisiä C-ohjelmointiohjelmistoja. Kaikki ohjelmointi oli mahdollista toteuttaa entuudestaan tutuilla ohjelmankehitysympäristöillä.

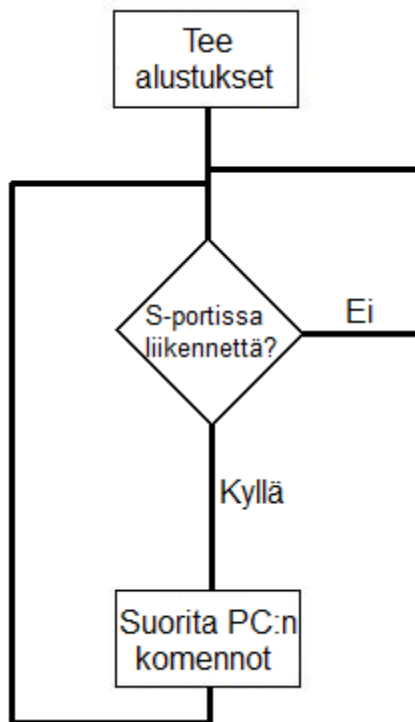
### 10.1 Mittalaitteen ohjelma

Mittalaitteen prosessorin ohjelmointi suoritettiin C-kielellä ja sen kirjoittamiseen käytettiin AVRStudio4:ää, joka on Atmelin AVR-laitteiden ohjelmointiympäristö. Lisäksi PC:llä täytyi olla asennettuna WinAVR, joka sisältää kääntäjän C-kielelle sekä työkalun (avrdude) ohjelman siirtämiseen AVR-laitteelle.

Ohjelmassa sovellettiin valmiita Internetistä saatavia anturin toimintoihin liittyviä kirjastofunktioita sekä koulussa opiskeltuja sulautetun laitteen ja PC:n välisen tietoliikenteen mahdollistavia funktioita. Kaikkia väylätekniikkaan ja anturin toimintaan liittyviä aiemmissa luvuissa esiteltyjä teorioita on käytetty hyväksi mittalaitteen ohjelmoinnissa. Mittalaitteen ohjelman päätehtävät ovat:

- sarjaportin käyttö
- kellonajan ylläpito
- lämpötilojen mittausrutiinit
- datan lähetys PC:lle.

Mikrokontrolleri toimii järjestelmässä isäntänä antureille. Toisaalta se toimii samalla myös orjana PC:lle. Mikrokontrollerin on siis kyettävä toimimaan kahteen suuntaan; se komentaa antureita tekemään mittaukset ajastetusti sekä kuuntelee samalla PC:ltä mahdollisesti tulevia käskyjä. PC:n käskyjen "odotteluosio" toimii ikisilmukassa, jossa tehdään tarvittavat rutiinit ainoastaan, mikäli sarjaportissa havaitaan merkki tai useita merkkejä. PC:ltä tulevat komennot liittyvät joko ajan asettamiseen tai lämpötiladatan keräämiseen. Toinen oma osionsa on timerin keskeytyspalveluohjelma, josta hypätään suorittamaan tietyt käskyt vain ajastinkeskeytyksen tapahtuessa. Ohjelman raamit voidaan esittää kahtena lohkokaaavana kuvioiden 25 ja 26 esittämällä tavoin.



KUVIO 25. Mittalaitteen toiminta lohkokaaviona

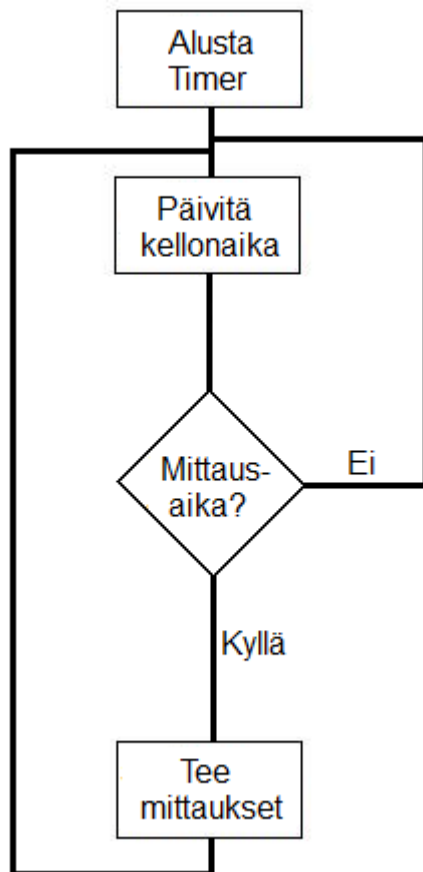
Yllä olevan lohkokaavion (kuvio 25) mukaisesti mittalaitteen ohjelmassa käydään koko ajan tarkastamassa, onko sarjaliikenneportissa liikennettä. Mikäli liikennettä on, mennään suorittamaan PC:n komentoja, jotka tässä tapauksessa ovat PC:n kellonajan vastaanottaminen sekä mittaustietojen lähettäminen. Mikäli portissa ei ole liikennettä, jatketaan merkkien odottamista. Main-funktion koodi vastaa kuvion 25 lohkokaaviota.

```

int main (void)
{
    USART_Init(207);           // alustetaan USART
    InitTimer();               // alustetaan timer tuottamaan keskeytys
    for(;;)                    // luuppi
    {
        if (IsCharAvailable() ) // tarkastetaan onko portissa liikennettä
        {
            HandleCommand();    // mennään suorittamaan PC:n komennot
        }
    }
}
  
```

Edellä HandleCommand()-funktio on rakennettu käyttäen case-rakennetta, jossa PC:ltä saadun komennon mukaan joko lähetetään mittausdataa tai asetetaan kellon-aika oikeaksi.

Tarkastellaan seuraavaksi laitteen ajastustoimintaa.



KUVIO 26. Ajastimen ja siihen liittyvien tapahtumien toiminta lohkokaaavana

Kuviossa 26 esitetään ajastimen toiminta ja sen laukaisemat tapahtumat. Ajastin alustetaan, jonka jälkeen alustuksessa määritettyjen parametrien mukaisesti tapahtuu ajastinkeskeytys tietyin väliajoin. Tässä määritetty väli on yksi millisekunti. Aina keskeytyksen tapahtuessa kasvatetaan muuttujaa yhdellä, jolloin tuhat keskeytystä on yhden sekunnin mittainen. Näin on saatu luotua sekuntikello. Ohjelmaan on määritetty mittausväliksi yksi tunti, joten koko ajan päivittyviä aikamuuttujia vertaillen ohjelma osaa mennä suorittamaan mittaukset tasatunnein.

Ajastimen funktiossa pidetään kellonaikaa yllä UpdateTime()-funktion kautta. Siellä tietyt kellonajat ohjaavat mittausfunktioon.

```

ISR(TIMER1_COMPA_vect)           // Timer1:n keskeytysaliohjelma
{
    msFromReset += 1;             // kasvatetaan millisekunteja
    if ((msFromReset % 1000) == 0) // jos 1000 millisekuntia täynnä...
        UpdateTime();            //...päivitetään aikaa (sekunteja)
}

```

Mittausfunktiossa aloitetaan ensin mittaus `StartMeasurement()`-funktioilla, jonka parametreina voidaan määrittää antureiden ulkoisen jännitelähteen käyttö. Sen jälkeen on pidettävä 750 mS:n tauko. Lopuksi jokaisen anturin lämpötilat luetaan `ReadTemperature()`-funktioilla, jonka parametreiksi annetaan antureiden yksilölliset tunnistenumerot sekä lämpötilataulukon tunti- ja anturikohtaiset solut. Näin lämpötilat luetaan kultakin anturilta taulukon oikeaan paikkaan. Taulukosta on siten helppo hakea lämpötilat PC:lle.

```

void DoMeasure (void)
{
    StartMeasurement ();           // author: Martin Thomas
    _delay_ms (750);              // pidetään tauko vaatimusten
                                   // mukaan

    ReadTemperature (SensorID1, temperature[1][hour]); // author: Martin Thomas
    ReadTemperature (SensorID2, temperature[1][hour]);
    ...
}

```

Mittautulokset tallennetaan mikrokontrollerin SRAM-muistiin 24-paikkaiseen lämpötilataulukkoon. Yksi mittausjakso kestää kokonaisuudessaan 7...10 sekuntia (riippuen käytettyjen sensoreiden määrästä), ja koska mittausrutiinit suoritetaan ajastinkeskeytysfunktion sisällä, tulee kellon aikaan joka kerta juuri edellä mainittu viive. Tämä tekee vuorokaudessa yhteensä min.  $7\text{ s} \cdot 24 = 2\text{ min } 48\text{ s}$  ja/tai max.  $10\text{ s} \cdot 24 = 4\text{ min}$  viivettä. Sillä ei sinänsä ole kuitenkaan merkittävää vaikutusta, koska mikrokontrollerin kellonaika asetetaan joka tunti ajastetusti PC:n kellon mukaiseksi. Näin viiveet nollataan 24 kertaa vuorokaudessa.

## 10.2 PC:n ohjelma

PC:n ohjelma toteutettiin C-kielellä ns. konsoliohjelmana käyttäen VisualStudio2008-ohjelmistoa. Sillä voidaan mm. kirjoittaa, kääntää ja debugata eri ohjelmointikieliä. Konsoliohjelmat eivät ole tarkalleen ottaen Windows-ohjelmia, vaan merkkipohjaisia ohjelmia, jotka suoritetaan DOS-ikkunassa. Ohjelman tehtävinä on asettaa mikrokontrollerin aika PC:n kellon mukaan, hakea mittausdata mikrokontrollerilta sekä tallentaa tiedot tekstitiedostoihin päivämäärän mukaan. Ohjelman päätehtävät ovat seuraavat:

- aikamuuttujien alustus ja käyttö
- sarjaportin käyttö ja tietoliikenteen isännöinti
- datan haku ja mikrokontrollerin kellonajan asetus
- haetun datan käsittely
- tiedostojen käsittely ja mittausdatan tallennus.

### 10.2.1 Ohjelman runko

Ohjelmarunko on kirjoitettu main-funktioon.

```
int _tmain()
{
    date_and_time(); // Alustetaan aikamuuttujat

    printf("TemperatureHoover.exe...\n\n"); // Ohjelman nimi..

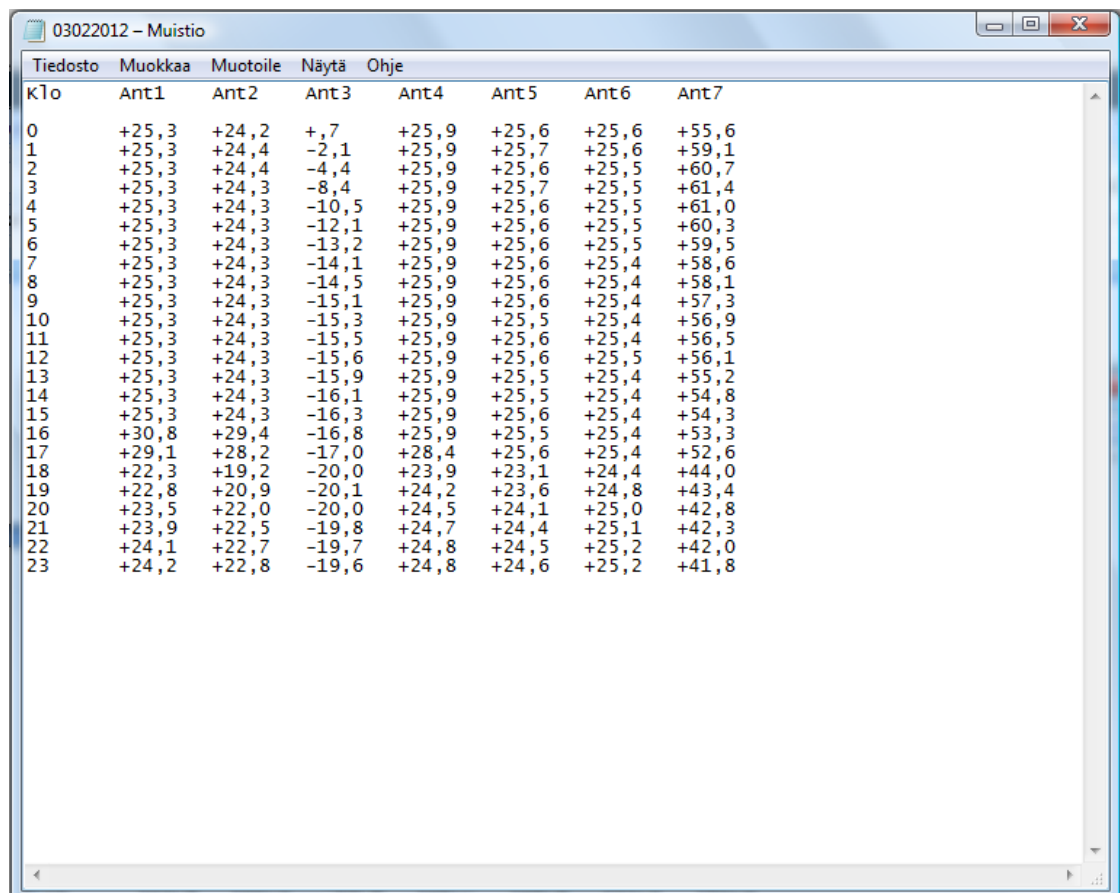
    if ( OpenConnection(2,COMM_TIMEOUT_USED) ) // Jos portti aukeaa..
    {
        Get_TodaysTemp(); // .. Haetaan dataa
        Get_YesterdaysTemp(); // .. Haetaan dataa
        SetTime(); // .. Asetaan aika
        printf("Haku suoritettu!");
        Sleep(5000);
    }
    else
        printf("COMX - portti varattu!!!");
        Sleep(5000);
    return 0;
}
```

Ensin alustetaan ohjelman aikamuuttujat (vuosi, kuu, päivä, tunti,...), jonka jälkeen yritetään avata sarjaliikenneyhteys mikrokontrolleriin. Mikäli se onnistuu, suoritetaan if-silmukassa oleva koodi. Kaksi ensimmäistä funktiota ovat lämpötiladatan hakua varten. Ensimmäinen hakee sen hetkisen vuorokauden mittausdatan ja jälkimmäinen edellisen vuorokauden datan. Mikrokontrollerin SRAMilla on lämpötilataulukko, johon kunkin anturin ja kunkin tunnin kohdalle tallennetaan aina oikea lämpötila. Koska lämpötilataulukko on 24-paikkainen, siihen mahtuu vuorokauden mittauks tulokset. Oletetaan että mittaukset on aloitettu eilen klo 15, joten nyt taulukossa on tuntien 15–

23 kohdalla eilisen päivän mittaustulokset. Tämän päivän data tallentuu paikasta 0 alkaen. Get\_TodaysTemp-funktio hakee datan alkaen tunnista 0 ja päättyen sen hetkiseen tuntiin. Get\_YesterdaysTemp-funktio hakee datan alkaen tunnista, joka on tämän hetkinen tunti + 1, eli ensimmäisestä paikasta eilisten tulosten kohdalla. Tämä tarkoittaa sitä, että mittaustulokset tulee hakea viimeistään vuorokauden kuluessa mittausten aloittamisesta, koska uudet mittaustulokset alkavat tallentumaan taulukossa vanhojen päälle. Haku päättyy klo 23:een. Näiden funktioiden sisällä hoidetaan tiedostojen käsittely ja datan tallennus. Molemmat funktiot lukevat tekstitiedostosta viimeisimmän tallennetun mittaustunnin/datan, jolloin haku jatkuu sitä seuraavan tunnin kohdalta. Jälkimmäistä funktiota ei käytetä jatkuvasti, sillä PC on ajastettu tekemään haun joka tunti, eikä tällöin ole yleensä tarvetta ”paikata” puuttuvia mittaustietoja eilisen päivän osalta.

## 10.2.2 Datan tallennus

Mittalaitteelta haettu data tallennetaan tekstitiedostoihin, jotka nimetään automaattisesti päivämäärän mukaan. Tiedoston rakenne näkyy kuvassa 17.



Tiedosto	Muokkaa	Muotoile	Näytä	Ohje			
Klo	Ant1	Ant2	Ant3	Ant4	Ant5	Ant6	Ant7
0	+25,3	+24,2	+7	+25,9	+25,6	+25,6	+55,6
1	+25,3	+24,4	-2,1	+25,9	+25,7	+25,6	+59,1
2	+25,3	+24,4	-4,4	+25,9	+25,6	+25,5	+60,7
3	+25,3	+24,3	-8,4	+25,9	+25,7	+25,5	+61,4
4	+25,3	+24,3	-10,5	+25,9	+25,6	+25,5	+61,0
5	+25,3	+24,3	-12,1	+25,9	+25,6	+25,5	+60,3
6	+25,3	+24,3	-13,2	+25,9	+25,6	+25,5	+59,5
7	+25,3	+24,3	-14,1	+25,9	+25,6	+25,4	+58,6
8	+25,3	+24,3	-14,5	+25,9	+25,6	+25,4	+58,1
9	+25,3	+24,3	-15,1	+25,9	+25,6	+25,4	+57,3
10	+25,3	+24,3	-15,3	+25,9	+25,5	+25,4	+56,9
11	+25,3	+24,3	-15,5	+25,9	+25,6	+25,4	+56,5
12	+25,3	+24,3	-15,6	+25,9	+25,6	+25,5	+56,1
13	+25,3	+24,3	-15,9	+25,9	+25,5	+25,4	+55,2
14	+25,3	+24,3	-16,1	+25,9	+25,5	+25,4	+54,8
15	+25,3	+24,3	-16,3	+25,9	+25,6	+25,4	+54,3
16	+30,8	+29,4	-16,8	+25,9	+25,5	+25,4	+53,3
17	+29,1	+28,2	-17,0	+28,4	+25,6	+25,4	+52,6
18	+22,3	+19,2	-20,0	+23,9	+23,1	+24,4	+44,0
19	+22,8	+20,9	-20,1	+24,2	+23,6	+24,8	+43,4
20	+23,5	+22,0	-20,0	+24,5	+24,1	+25,0	+42,8
21	+23,9	+22,5	-19,8	+24,7	+24,4	+25,1	+42,3
22	+24,1	+22,7	-19,7	+24,8	+24,5	+25,2	+42,0
23	+24,2	+22,8	-19,6	+24,8	+24,6	+25,2	+41,8

KUVA 17. Lämpötilojen tallennustiedoston sisältö (Jesse Nissinen 2012)

Tiedoston nimenä on päivämäärä muodossa pppkkvvvv. Lämpötilat on taulukoitu tiedostoon tuntien ja antureiden mukaan.

### 10.3 Excel-käyttöliittymä

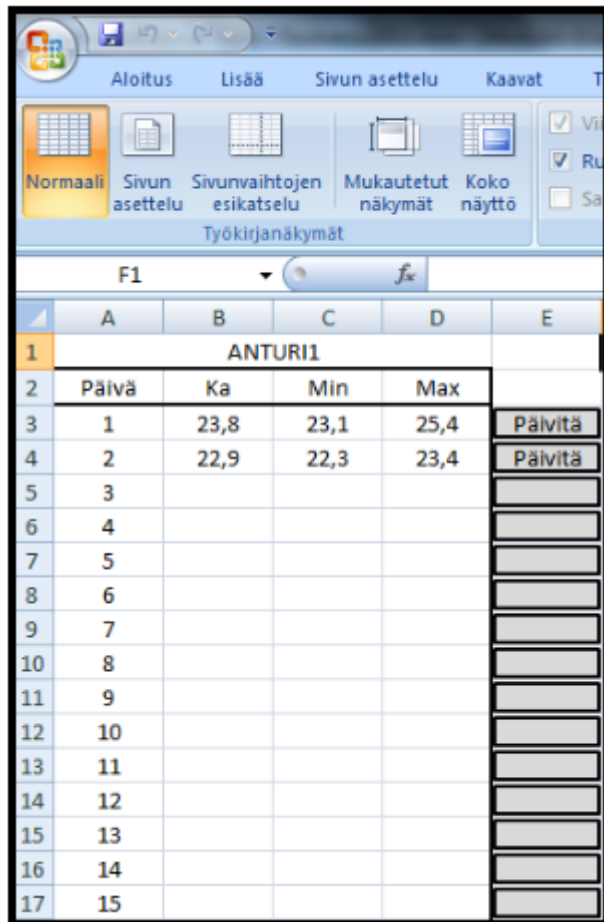
Excel-käyttöliittymän tarkoituksena on hakea data tekstitiedostoista ja laskea kunkin anturin lämpötiloista keskiarvot, minimi ja maksimit. Tämä kaikki toteutettiin makroilla, jotka suoritetaan makronappia painamalla. Makrot ovat Excelin VBA-editorin avulla luotavia ohjelmakoodin pätkiä. Lämpötilojen tallennuskansiossa on jokaiselle kuukaudelle omat alikansionsa, jonne on luotu jokaiselle anturille excel-tiedosto. Näin yhdessä Excel-tiedostossa on yhden anturin kuukauden jokaisen päivän lämpötilan keskiarvo, minimi ja maksimi. Lämpötiladatan tarkastelu on selkeää, kun kullekin mitauskohteelle on oma tiedostonsa.

Ohjelmallisesti Excel-ohjelma voidaan jakaa seuraaviin pääkohtiin:

- datan haku lämpötilatiedostoista
- tarvittavien laskutoimituksien suorittaminen haetusta datasta
- kuvaajien piirtäminen lasketuista luvuista

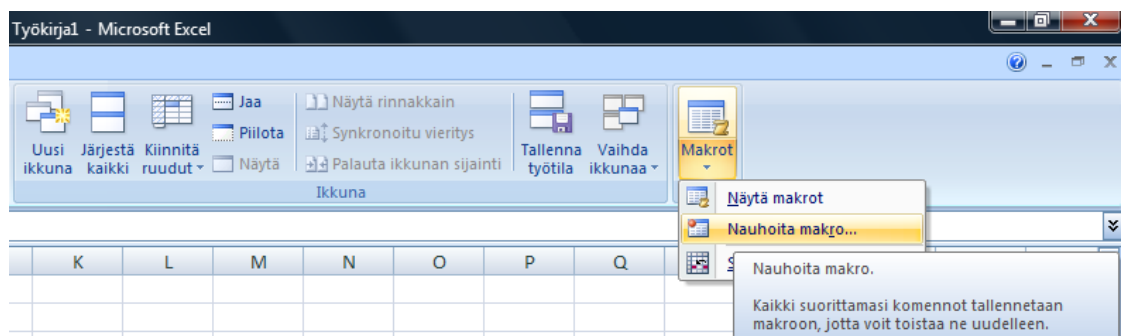
Kuvan 18 mukaisesti päivitä-nappia painamalla saadaan tekstitiedostoon tallennettua datasta laskettua halutut arvot. Luvut sijoittuvat soluihin päivien mukaan.





KUVA 18. Excel-käyttöliittymä

Ohjelmakoodia generoitiin melko paljon nauhoittamalla Excel-makroja. Makrojen nauhoitus onnistuu Excelissä siten, että näytä-välilehdeltä valitaan makrot ja nauhoita makro (kuva 19).



KUVA 19. Excel-makrojen nauhoitus

Tämän jälkeen Excel-taulukko voidaan tehdä melkein mitä tahansa toimenpiteitä ja kaikki nauhoitetaan VBA-editoriin toimintoja vastaavana ohjelmakoodina. Kun kaikki tarvittava on tehty, valitaan makrot-valikosta "lopetta nauhoittaminen". Koodin generointi loppuu tähän. Tämän jälkeen koodia on helppo muokata tarpeen mukaan edito-

riassa. Koska Excel-käyttöliittymän yksi tärkeimmistä tehtävistä oli hakea data kovalevylle tallennetuista lämpötilatiedoista, nauhoitettiin ensiksi yksinkertainen tietojen haku tekstitiedostosta. Myöhemmin siitä karsittiin turhaa valmiiksi generoitua koodia ja lisättiin tarpeellisia osia.

#### 10.4 Toiminnan automatisointi ja varmuuskopiointi

Lämpötilojen haku mittalaitteelta haluttiin automaattiseksi, joten PC:n ohjelma täytyi virittää käynnistymään itsenäisesti tiettyinä kellonaikoina. Yksinkertaisimmaksi tavaksi osoittautui Windowsin ajastettujen tehtävien (scheduled tasks) käyttö. Sen avulla mikä tahansa ohjelma voidaan asettaa käynnistymään tiettyjen tapahtumien liipaisemana. Näitä voivat olla esimerkiksi sisäänkirjautuminen Windowsiin, tietokoneen käynnistämien ja kellonajat. Tässä tapauksessa kellonajat olivat tarpeellisin parametri. Ajastus asetettiin toimimaan joka päivä tunnin välein, joten mittausjärjestelmän käyttäjän ei tarvitse olla jatkuvasti ajamassa ohjelmaa itse manuaalisesti. Näin lämpötiladata saadaan PC:n kovalevylle halutuin väliajoin automaattisesti ja mittalaitteen aika synkronoituu kerran tunnissa PC:n kellon kanssa.

Osaksi järjestelmää haluttiin varmuuskopiointiominaisuus. Tarkoituksena oli kopioida arkistoitu data verkkolevylle. Yhtälailla tiedostot haluttiin saada verkkolevyltä automaattisesti myös muille verkkoa käyttäville tietokoneille. Näin lämpötiloja olisi mahdollista seurata usealta tietokoneelta. Ohjelmallisesti varmuuskopiointi toteutettiin luomalla kaksi erilaista BAT-tiedostoa, joihin kirjoitettu ohjelmakoodi kopioi tiedostot paikasta toiseen. BAT on tiedostopääte MS-DOS:ssa suoritettaville komennoille (toimivat myös Windowsin komentorivin kautta). BAT-tiedosto luodaan tallentamalla tekstitiedosto .bat-päätteisenä. Tiedostoon voidaan kirjoittaa ohjelmakoodi rivi kerrallaan. Ohjelmaa ajettaessa avautuu MS-DOS-ikkuna, jossa koodin suoritus ja tehtävät toiminnot näkyvät. Nämäkin ohjelmat ajastettiin samoin, kuten aiemmassa kappaleessa kerrottiin, joten järjestelmään saatiin automaattinen varmuuskopiointi ja tiedostojen verkotus.

## 11 YHTEENVETO JA JATKOKEHITYSMAHDOLLISUUDET

Projektin lopputulos oli työn tilaajan mielestä onnistunut. Laitteet saatiin toimintavaroiksi ja koko järjestelmä toimivaksi ja helppokäyttöiseksi. Työ itsessään vaati paljon sulautettujen järjestelmien opettelua ja osaamista. Koko prosessi oli hyvin hyödyllinen ja opettavainen. Sulautetut järjestelmät ovat erittäin suuri osa elektroniikka-alaa, mikä vuoksi niiden tunteminen on hyvin tärkeää.

Koska kaikki on toiminut tähän asti hyvin, on laitteistoa mahdollista kehittää jatkossa esimerkiksi etävalvontajärjestelmäksi. Etävalvontajärjestelmä voitaisiin toteuttaa lisäämällä mittalaitteen mikrokontrollerin yhteyteen gsm-moduuli, joka lähettäisi tekstiviestejä lämpötilarajojen ylittyessä tai alittuessa. Nyt Excel-käyttöliittymään voidaan asettaa halutut raja-arvot eri mittauspisteiden lämpötiloille. Rajojen rikkoutuessa saadaan visuaalinen palaute päivitettäessä lämpötilataulukkoa. Gsm-moduulin mukanaan tuoma välitön palautteen saantimahdollisuus matkapuhelimeen toisi järjestelmälle paljon lisäarvoa. Mittalaite ei kuitenkaan ole ainoa järjestelmän osa, joka on synnyttänyt kehitysideoita. PC:n ohjelmistoa voisi laajentaa mm. toimimaan yhdessä sähköpostin kanssa. Ohjelma voisi lähettää tietoja laitteiston toiminnasta ja jopa kokonaisia lämpötilaraportteja usealle käyttäjälle.



## LÄHTEET

Arduino. 2011. *Arduino Duemilanove*. [verkkodokumentti]. [viitattu 15.6.2011]. Saatavissa: <http://arduino.cc/en/Main/arduinoBoardDuemilanove>

Arduino. 2012. *DigitalPins*. [verkkodokumentti]. [viitattu 12.2.2012]. Saatavissa: <http://arduino.cc/en/Tutorial/DigitalPins>

Bebek Electronic. 2012. *LAN MODULAR 02 8 PIN*. Yrityksen internetsivut. [viitattu 22.1.2012]. Saatavissa: <http://www.bebek.fi/kauppa/index2.php>

Durda, F. 1996. *Serial and UART Tutorial*. [verkkodokumentti]. [viitattu 1.3.2012]. Saatavissa: <http://www.freebsd.org/doc/en/articles/serial-uart/>

Koskinen, J. 2004. *Mikrotietokonetekniikka. Sulautetut järjestelmät*. Keuruu: Otavan kirjapaino Oy.

Linke, B. 2008. *Overview of 1-wire Technology and Its Use*. [verkkodokumentti]. [viitattu 14.7.2011]. Saatavissa: <http://www.maxim-ic.com/app-notes/index.mvp/id/1796>

Made-in-China.com. 2012. *UTP Cat5e Patch Cable*. Yrityksen internetsivut. [viitattu 2.1.2012] Saatavissa: <http://dongzhuocable.en.made-in-china.com/product/wqXQRcAMnoVg/China-UTP-Cat5e-Patch-Cable.html>

Maxim Integrated Products. 2009. *Comparison of the DS18B20 and DS18S20 1-Wire Digital Thermometers*. [Pdf-dokumentti]. [viitattu 12.12.2011]. Saatavissa: <http://www.maxim-ic.com/app-notes/index.mvp/id/4377>

Maxim Integrated Products. 2008. *Guidelines for Reliable Long Line 1-Wire Networks*. [verkkodokumentti]. [viitattu 22.5.2011]. Saatavissa: <http://www.maxim-ic.com/app-notes/index.mvp/id/148>

Maxim Integrated Products. 2010. *High-Precision 1-Wire Digital Thermometer*. [Pdf-dokumentti]. [viitattu 22.5.2011]. Saatavissa:

<http://datasheets.maxim-ic.com/en/ds/DS18S20.pdf>

Memmert. 2012. *Cooled Incubator ICP-500*. Yrityksen internetsivut. [viitattu 12.1.2012]. Saatavissa:

<http://www.memmert.com/products/cooled-incubator/cooled-incubators-models/models/ICP-500/>

Sanyo. 2012. *Product Family Brochure*. [Pdf-dokumentti]. [viitattu 15.1.2012]. Saatavissa:

<http://www.sanyobiomedical.com/assets/documents/brochure/MIRSeriesIncubatorBrochure.pdf>

Maksimainen, V. 2009. *Sarjaportin käyttö tiedonsiirtoon PC:lle*. Kuopio: Savonia-ammattikorkeakoulu. Opetusmateriaali.

Springbok Digitronics. 2004. *1-Wire Design Guide v1.0*. [Pdf-dokumentti]. [viitattu 21.7.2011]. Saatavissa:

<http://www.1wire.org/Files/Articles/1-Wire-Design%20Guide%20v1.0.pdf>

Vahtera, P. 2008. *Ajastin/Laskuri – Timer/Counter*. [verkkodokumentti]. [viitattu 18.8.2011]. Saatavissa:

<http://www.scribd.com/doc/36922351/6-6-1-AVR-rauta-Timer-ohjelmointia>

Vitronia.com. 2012. *DS18S20*. Yrityksen internetsivut. [viitattu 11.1.2012] Saatavissa:

[http://www.vitronia.com/shop/product.php?id\\_product=147](http://www.vitronia.com/shop/product.php?id_product=147)

Wikipedia. 2012. *Parikaapeli*. [verkkodokumentti]. [viitattu 16.10.2011]. Saatavissa:

<http://fi.wikipedia.org/wiki/Parikaapeli>

Wikipedia. 2011a. *RS232*. [verkkodokumentti]. [viitattu 18.11.2011]. Saatavissa:

<http://fi.wikipedia.org/wiki/RS-232>

Wikipedia. 2011b. *Tarkistussumma*. [verkkodokumentti]. [viitattu 2.12.2011]. Saatavissa:

<http://fi.wikipedia.org/wiki/Tarkistussumma>







## MITTAUSLAITTEISTON KÄYTTÖOHJEET

Tässä liitteessä on mittauslaitteiston käyttöohjeet, jotka on pyritty tekemään mahdollisimman selkokielisiksi.

### MITTAUSLAITTEISTO

Tämä lämpötilanseurantajärjestelmä on toteutettu kahdella mittalaitteella ja niihin kytketyillä anturiverkoilla sekä kannettavalla PC-tietokoneella (kuva 1). Anturiverkot on jaettu fyysisesti siten, että toinen verkko mittaa lämpötilat laboratorion lämpökaapeista ja toinen käytävän varrella olevista kylmäsäilytystiloista. Mittalaitteet tekevät mittaukset ajastetusti kustakin mittauspisteestä ja PC hakee tietyin väliajoin nämä mittaustulokset kovalevylle talteen.

### ALKUVALMISTELUT

Aluksi tulee varmistua siitä, että kaikki anturit ovat kytkettyinä haaroittimien kautta mittausverkkoon ja itse verkko on kytketty mittalaitteeseen verkkoliittimen kautta. Tämän jälkeen mittalaite voidaan kytkeä PC:hen USB-johdolla. Kytkennät tehdään alla olevan kuvan mukaisesti. Virtajohdon (kolmas ohut musta johto oikealla alareunassa) kytkeminen mittalaitteeseen ei ole välttämätöntä, sillä laite saa virtansa USB-väylän kautta.



KUVA 1. Mittalaitteeseen tehtävät liitännät

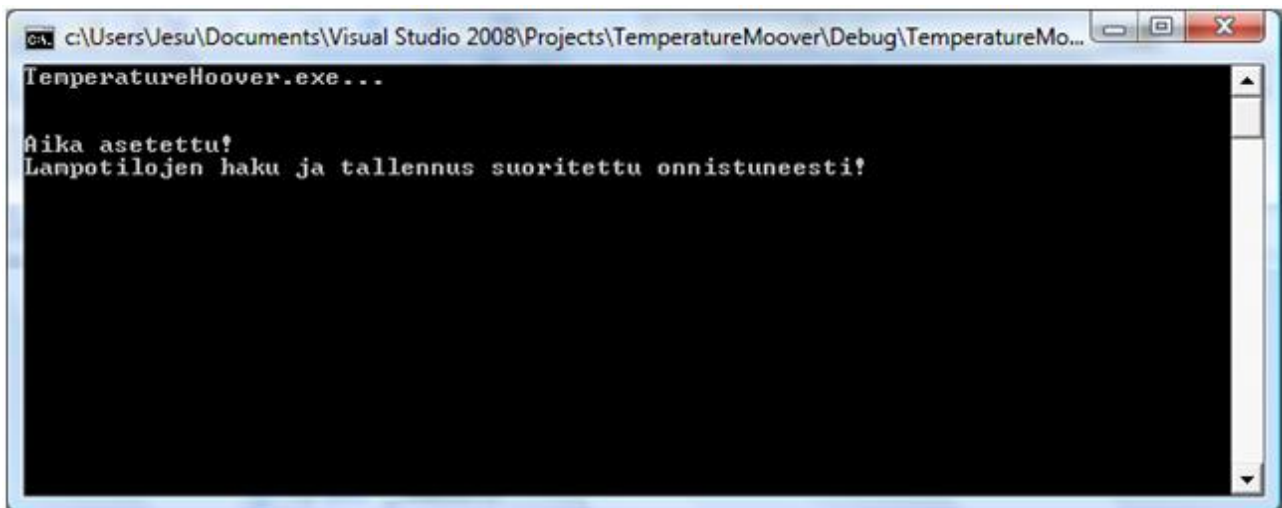
Kun kytkennät on tehty oikein ja mittalaite on liitetty USB-johdolla PC:hen, korttiin syttyy muutaman sekunnin kuluttua vihreä valo. Tämä on merkinä siitä, että laitteeseen tulee virta ja että kaikki anturit on havaittu ja ne ovat toimintavalmiudessa. ON ÄÄRIMMÄISEN TÄRKEÄÄ SUORITTAA OHJELMA **TEMPERATUREHOOVER.EXE** MAHDOLLISIMMAN PIAN SEN JÄLKEEN, KUN VIHREÄ VALO ON SYTTYNUT! Tämä siksi, että laite toimii heti alusta lähtien oikean kellon ajan mukaan.

## MITTAUKSET

Laite on ohjelmoitu suorittamaan mittaukset tasatunnein tunnin välein. Tällöin uusimmat mittaustulokset ovat valmiina laitteen muistissa käytännössä heti tunnin vaihtumisen jälkeen, jolloin ne ovat myös haettavissa PC:lle. Mittaustulokset säilyvät mittalaitteen muistissa 24 tuntia, joten hakuvälin tulee olla mielellään alle 24 tuntia.

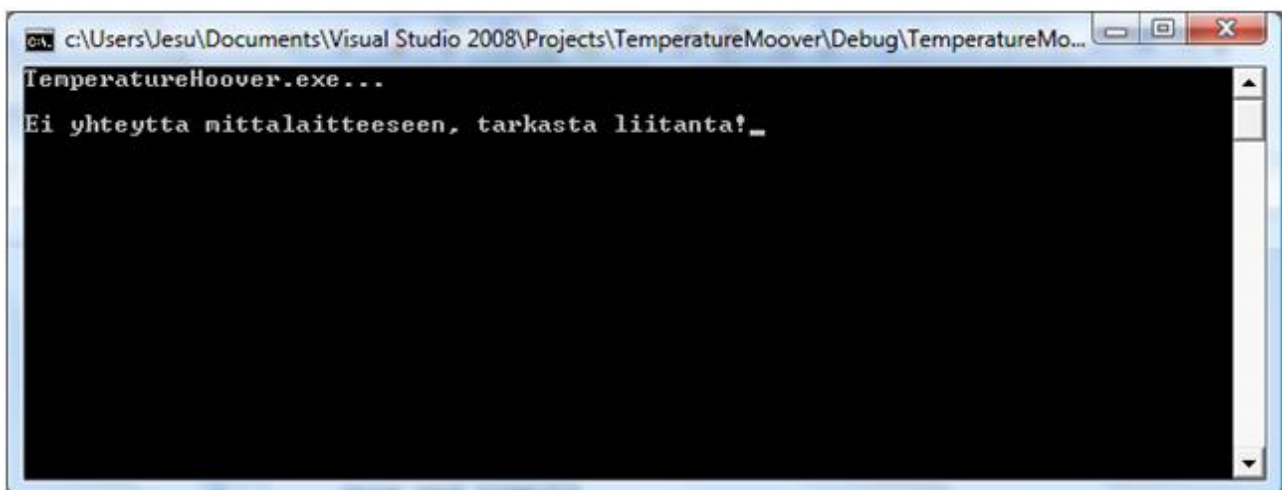
## LÄMPÖTILADATAN HAKEMINEN PC:LLE

Mittaustulokset haetaan PC:lle ajamalla työpöydällä oleva TemperatureHoover(\_Cold).exe. Tämä tehdään myös täysin automaattisesti kerran tunnissa. Haun onnistuessa avautuva ohjelmaikkuna on esitetty kuvassa 2.



KUVA 2. Onnistuneesta lämpötilanhausta ilmoitetaan ohjelmaikkunassa

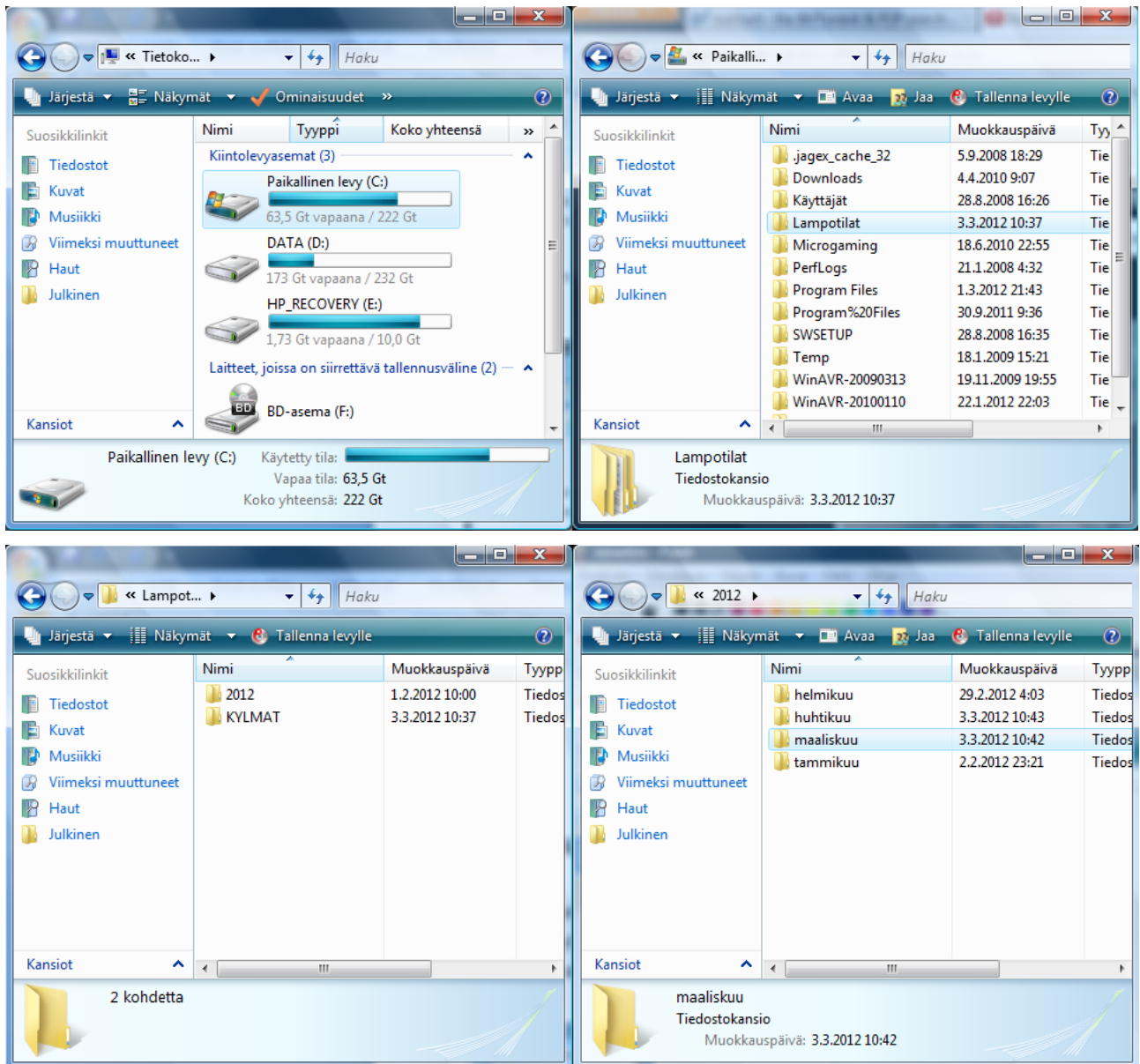
Mikäli PC ei saa yhteyttä mittalaitteeseen, ilmoittaa ohjelma siitä kuvan 3 mukaisesti.



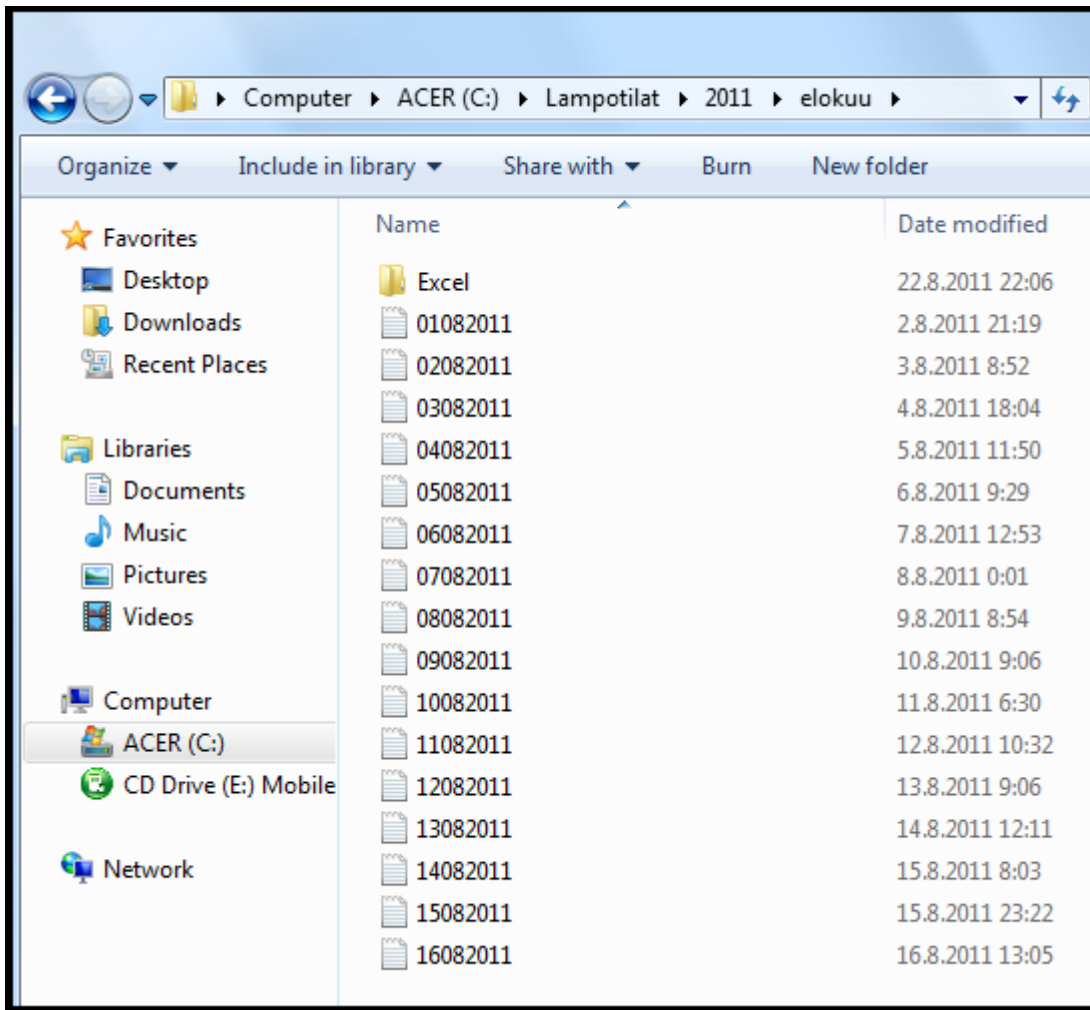
KUVA 3. Epäonnistuneesta lämpötilanhausta seuraa ilmoitus

## DATAN ARKISTOINTI

Mittausdata arkistoidaan automaattisesti **C-levyllä** olevaan **Lampotilat-kansioon**. (C:\Lampotilat). Kansiota löytyy alikansiot vuosien sekä kuukausien mukaan. Kylmien mittauspisteiden data on Lampotilat-kansion KYLMAT-kansiossa (C:\Lampotilat\KYLMAT). Kuva 4 havainnollistaa sen, minne mittautustiedot arkistoidaan.



KUVA 4. Mittautustiedot tallennetaan kuukausikohtaisiin kansioihin



KUVA 5. Mittausdata tallentuu tekstitiedostoihin päivämäärien mukaan

Mittausdata on tekstitiedostoissa päivämäärien mukaan (kuva 5). Esimerkiksi vuoden 2011 elokuun 16. päivän mittautiedot löytyvät tiedostosta 16082011.

Tekstitiedostoon ei tulisi tehdä muutoksia itse, sillä tekstitiedoston sisällön olisi säilyttävä alkuperäisessä muodossaan sen vuoksi, että ohjelma osaa tallentaa aina kunkin anturin lämpötilatiedot myöhemmin oikeaan paikkaan. Tiedoston avaaminen/sulkeminen ja tietojen tarkastelu on tietenkin sallittua ja suotavaa. Tiedosto kannattaa aina tallentaa ennen sulkemistaan.

Kunkin kuukauden kansiossa on lisäksi Excel-alikansio, jossa on kullekin anturille oma Excel-tiedostonsa. Ne näyttävät seuraavanlaisilta (jokaisen taulukon yläosassa lukee anturin numero/nimi).

	A	B	C	D	E
1	ANTURI1				
2	Päivä	Ka	Min	Max	
3	1	23,8	23,1	25,4	Päivitä
4	2	22,9	22,3	23,4	Päivitä
5	3				
6	4				
7	5				
8	6				
9	7				
10	8				
11	9				
12	10				
13	11				
14	12				
15	13				
16	14				
17	15				
18	16				
19	17				
20	18				
21	19				
22	20				
23	21				
24	22				
25	23				
26	24				
27	25				
28	26				
29	27				
30	28				
31	29				
32	30				
33	31				

KUVA 6. Järjestelmän Excel-käyttöliittymä

Kuvassa 6 näkyvää päivitä-nappia painamalla Excel hakee ja laskee automaattisesti kullekin päivälle mittaustuloksista keskiarvon, minimin ja maksimin.

## MAHDOLLISET VIKATILANTEET

Mittalaitteessa on virheen tunnistus, joka ilmoittaa mahdollisista virhetilanteista punaisella merkki-valolla. Punainen valo palaa tällöin vihreän valon vieressä ja vihreä valo sammuu.

Mahdollisia virhetilanteita mittalaitteessa:

- Alkuvalmisteluiden yhteydessä verkkoa ei ole kytketty mittalaitteeseen
- Alkuvalmisteluiden yhteydessä joku antureista on jäänyt kytkemättä
- Mittauksen aikaan laite havaitsee jonkun anturin tai verkon olevan irti (tällöin tekstitiedoston tulostuu merkki "-" kaikkien niiden antureiden/kellonajan kohdalle, jolta mittaustulos puuttuu)

Tällöin laitteesta tulee katkaista virta pois ottamalla USB-johto irti (myös musta virtajohto mikäli se on kytketty) ja tarkistaa kaikki kytkennät. Jos jokin johto tai anturi on irti, se tulee liittää paikalleen ja kytkeä korttiin taas virrat päälle. Sen jälkeen ajetaan ohjelma TemperatureHoover.exe/TemperatureHoover\_Cold.exe, riippuen kumpi mittalaitteista on kyseessä.

PC:n puolelta aiheutuvia virhetilanteita ovat seuraavat:

- USB-johto ei ole kiinni mittalaitteessa
- USB-johto ei ole kiinni PC:ssä
- Ohjauspaneelin asetuksissa mittalaitteen com-portti on jostain syystä vaihtunut toiseksi (epätodennäköistä)

Edellä mainituissa tilanteissa ohjelma ilmoittaa virheestä kuvan 3 mukaisesti.

Ongelmista kaksi ensimmäistä poistuvat yksinkertaisesti liittämällä USB-johto sekä PC:hen että mittalaitteeseen. Mikäli se ei auta, on ongelma com-portin asetuksissa. Lämpökaappien mittalaite käyttää PC:n com1-porttia ja kylmien lämpötilojen mittalaite com2-porttia. Tällöin tilanne voidaan tarkastaa ja muokata ohjauspaneelin laitehallinnasta. Ohjeet tälle toimenpiteelle ovat seuraavat:

1. Kytke mittalaite normaalisti USB-johdolla kiinni PC:hen.
2. Avaa käynnistä valikosta **Ohjauspaneeli** (Käynnistä -> Ohjauspaneeli).
3. Seuraavaksi avautuneesta näkymästä klikkaa auki **"System"**.
4. Avaa välilehti **"Hardware"** ja sieltä klikkaa auki **"Devices and Printers"**.
5. Aukeavassa ikkunassa on pudotusvalikko **"com-ports"**.
6. Irrota halutun mittalaitteen USB-johto PC:stä, jolloin "com-ports"-valikko elää.

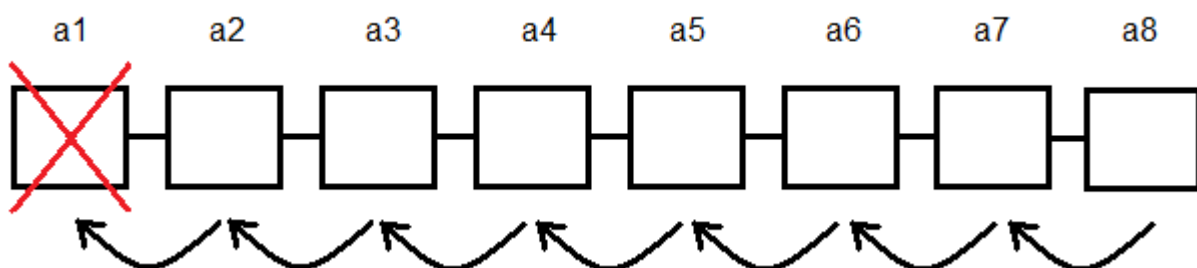
7. Kiinnitä USB-johto takaisin, jolloin ”com-ports”-valikkoon tulee uutena laitteena ko. mittalaitte.
8. Klikkaa hiiren oikealla näppäimellä ilmestynyttä laitetta ja avaa valikosta ”**properties**”.
9. Avautuvan ikkuna välilehdeltä löytyy kohta ”**Advanced**”, avaa se.
10. Valitse pudotusvalikosta portti 1 tai 2 riippuen siitä, kumpi mittalaite on kyseessä ja muista painaa **apply/ok** poistuessasi ikkunasta.
11. Aja mittalaitteen ohjelma työpöydältä varmistaaksesi toiminta sekä asettaaksesi kellonaika.

## VARMUUSKOPIOINTI

Järjestelmässä on varmuuskopiointitoiminto, joka tallentaa laboratorion päätietokoneeseen kerätyn mittausdatan verkkolevyille (J-levy). Toimintoa voidaan tarpeen mukaan käyttää myös toiseen suuntaan; tiedot saadaan automaattisesti tallennettua verkkolevyiltä muillekin verkkoa käyttäville tietokoneille, joten lämpötiloja voidaan seurata useammalla tietokoneella.

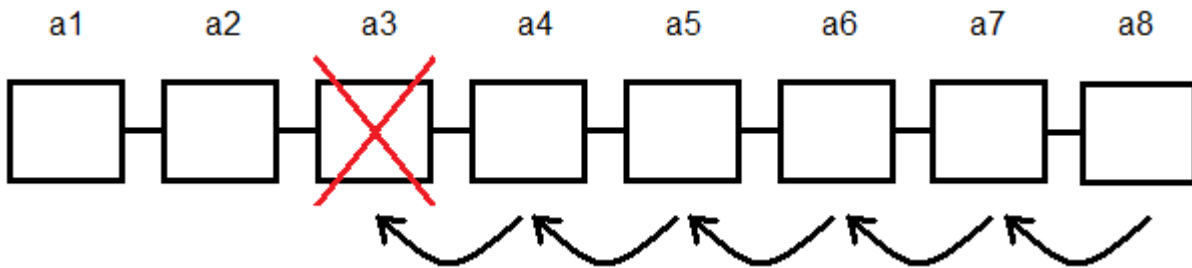
## KORJAUSTOIMET – ANTURIN VAIHTAMINEN

Anturin vikaantuessa se voidaan vaihtaa uuteen irrottamalla entinen anturi haaroittimesta ja korvaamalla se uudella anturilla. Tässä tilanteessa tulee ottaa huomioon vaihdettavan anturin/antureiden järjestysnumerot. Anturit on numeroitu valmiiksi niiden yksilöllisten tunnistenumeroiden mukaisessa järjestyksessä numerosta 1 alkaen. Esimerkkinä voidaan ajatella tilanne, jossa huomataan anturin 1 olevan rikki. Tilanne näkyy konkreettisesti lämpötilatiedostossa, jossa anturin 1 kohdalle tulostuu merkki ”-”. Oletetaan, että antureita verkossa on seitsemän. Anturit ovat verkossa siten, että 1. mittauspisteessä on anturi nro 1; 2.mittauspisteessä on anturi nro 2 jne. Anturin nro 1 mennessä rikki, anturi nro 2 ottaa sen paikan. Tällöin anturi nro 3 ottaa anturi nro 2 paikan jne. Eli jonkun anturin vikaantuessa seuraavaksi lähinnä oleva anturi tulee sen tilalle ja viimeiseltä paikalta vaihdetun anturin tilalle laitetaan uusi vara-anturi (kuvio 1). Vara-anturit on numeroitu samoin kuten muutkin anturit, eli ensimmäinen vara-anturi olisi tässä tapauksessa anturi nro 8, seuraava vara-anturi on anturi nro 9 jne.



KUVIO 1. Vikaantuneen anturin tilalle voidaan vaihtaa toimiva anturi

Seuraavassa esimerkissä (kuvio 2) anturi nro 3 on rikki, joten anturit nro 1 ja 2 saavat jäädä paikalleen. "Anturiletkaa" vedetään yhtä paikkaa taaksepäin alkaen anturista nro 4.



KUVIO 2. Anturin vikaantuessa se korvataan järjestyksessä seuraavana olevalla anturilla



## BOM

BOM (Bill Of Materials) sisältää tiedot laitteistoon käytetyistä osista ja komponenteista.

Tilaaja					
Työterveyslaitos Kuopio					
Tunniste	Osa	Tyyppi	Määrä	Toimittaja	Hinta/€
Laitteisto					
Arduino Duemilanove	Mikrokontrolleri	Atmega328P	2	Robomaa	45,8
Dallas DS18S20	Lämpötila-anturi	TO92-kotelo	12	Farnell	64,56
Kaapelit					
USB	Aktiivinen USB-johto	10m	1	Kauppasatama	13
CAT5	Parikaapeli	UTP 0,6m	12	Elfa	16,56
		UTP 2m	6	Elfa	17,58
		UTP 5m	5	Elfa	28,5
		UTP 7,5m	2	Elfa	17,4
Haaroittimet					
LAN MODULAR 01 8 PIN	Cat5-kaapelin jatkopala	RJ45	3	Bebek	6
LAN MODULAR 02 8 PIN	Verkon haaroitin	RJ45	12	Bebek	36
Virtalähde					
Arduinon virtalähde	DC Verkkolaite	12V 0.6A 2.1mm	2	Nadenex	15
Kondensaattorit					
C1	Elektrol.kondensaattori		2	Robomaa	0,3
Vastukset					
R1		4,7k	2	Robomaa	0,2
R2,R3		100	4	Robomaa	0,4
Diodit					
D1	LED-lamppu	Punainen	2	Robomaa	2
D2	LED-lamppu	Vihreä	2	Robomaa	2
Muut					
Duemilanoven kotelo	Suojakotelo		2	Robomaa	9,95
Hyppylangat	KytKentälanka			Robomaa	7,6
Kutistemuovi	Anturin suojamuovi	12,7mm; 1,2m	1	Vekoy	2,52
		3mm; 1,2m	1	Vekoy	2,32
Peitelista	Kaapeleiden peitelista	1cm; 2m	15	Prisma	30
				<b>Hinta yhteensä</b>	<b>317,69</b>



